Amazon Redshift

Management Guide

API Version 2012-12-01

Amazon Redshift: Management Guide

Copyright © 2015 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, AWS CloudTrail, AWS CodeDeploy, Amazon Cognito, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Amazon Kinesis, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC, and Amazon WorkDocs. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon Redshift? ... 1
- Are You a First-Time Amazon Redshift User? ... 1
- Amazon Redshift Management Overview ... 2
  - Cluster Management ... 2
  - Cluster Access and Security ... 2
  - Monitoring Clusters ... 3
  - Databases ... 4

Clusters ... 5
- Overview ... 5
- About Clusters and Nodes ... 6
  - Determining the Number of Nodes ... 7
- Resizing a Cluster ... 8
- Supported Platforms to Launch Your Cluster ... 9
  - EC2-Classic Platform ... 9
  - EC2-VPC Platform ... 9
  - Choose a Platform ... 10
- Regions and Availability Zone Considerations ... 10
- Maintenance Windows ... 10
- Default Disk Space Alarm ... 11
- Renaming Clusters ... 11
- Shutting Down and Deleting Clusters ... 12
- Managing Clusters Using the Console ... 13
  - Creating a Cluster ... 14
  - Modifying a Cluster ... 20
  - Shutting Down or Deleting a Cluster ... 22
  - Rebooting a Cluster ... 23
  - Resizing a Cluster ... 24
  - Getting Information About Cluster Configuration ... 26
  - Getting an Overview of Cluster Status ... 26
  - Taking a Snapshot of a Cluster ... 27
  - Editing the Default Disk Space Alarm ... 28
  - Working with Cluster Performance Data ... 29
- Managing Clusters Using the AWS SDK for Java ... 29
- Manage Clusters Using the Amazon Redshift CLI and API ... 31
- Managing Clusters in an Amazon Virtual Private Cloud (VPC) ... 32
  - Overview ... 32
  - Creating a Cluster in a VPC ... 33
  - Managing VPC Security Groups for a Cluster ... 35
  - Cluster Subnet Groups ... 36

Security Groups ... 43
- Overview ... 43
- Managing Cluster Security Groups Using the Console ... 44
  - Creating a Cluster Security Group ... 45
  - Tagging a Cluster Security Group ... 46
  - Managing Ingress Rules for a Cluster Security Group ... 47
  - Revoking Ingress Rules for a Cluster Security Group ... 49
  - Tagging Ingress Rules for a Cluster Security Group ... 49
  - Deleting a Cluster Security Group ... 51
  - Associating a Cluster Security Group with a Cluster ... 52
- Managing Cluster Security Groups Using the AWS SDK for Java ... 52
- Manage Cluster Security Groups Using the Amazon Redshift CLI and API ... 55

Parameter Groups ... 56
- Overview ... 56
- Configuring the Maximum Size of a Cursor Result Set ... 57
- Configuring Workload Management ... 59

Configuring the wlm_json_configuration Parameter ... 69
Managing Parameter Groups Using the Console ... 72
Creating a Parameter Group ... 73
Modifying a Parameter Group ... 74
Deleting a Parameter Group ... 76
Associating a Parameter Group with a Cluster ... 77
Managing Parameter Groups Using the AWS SDK for Java ... 77
Managing Parameter Groups Using the Amazon Redshift CLI and API ... 80
Snapshots ... 82
Overview ... 82
Automated Snapshots ... 82
Manual Snapshots ... 83
Copying Snapshots to Another Region ... 83
Restoring a Cluster from a Snapshot ... 84
Sharing Snapshots ... 84
Managing Snapshots Using the Console ... 86
Creating a Manual Snapshot ... 87
Deleting a Manual Snapshot ... 88
Copying an Automated Snapshot ... 88
Restoring a Cluster from a Snapshot ... 89
Sharing a Cluster Snapshot ... 91
Configuring Cross-Region Snapshot Copy ... 92
Modifying the Retention Period for Cross-Region Snapshot Copy ... 93
Disabling Cross-Region Snapshot Copy ... 94
Managing Snapshots Using the AWS SDK for Java ... 94
Managing Snapshots Using the Amazon Redshift CLI and API ... 97
Database Encryption ... 98
Overview ... 98
Hardware Security Modules ... 99
Configuring HSM Using the Amazon Redshift Console ... 99
Configuring HSM Using the Amazon Redshift CLI and API ... 105
Rotating Encryption Keys ... 105
Rotating Encryption Keys Using the Amazon Redshift Console ... 105
Rotating Encryption Keys Using the Amazon Redshift CLI and API ... 106
Purchasing Reserved Nodes ... 107
Overview ... 107
What Are Reserved Node Offerings? ... 107
Purchasing a Reserved Node Offering with the Console ... 108
Listing Reserved Node Offerings ... 109
Purchasing Reserved Node Offerings ... 109
Purchasing a Reserved Node Offering Using Java ... 111
Purchasing a Reserved Node Offering Using the Amazon Redshift CLI and API ... 114
Controlling Access to Amazon Redshift Resources ... 115
Overview of Access to Amazon Redshift Resources ... 115
Using IAM Users, Groups, and Policies for Cluster Management ... 116
AWS Managed Policies for Amazon Redshift ... 116
Creating an IAM Group and Users ... 116
IAM Policy Elements for Amazon Redshift ... 117
IAM Policy Actions for Amazon Redshift ... 118
IAM Policy Resources for Amazon Redshift ... 118
Constructing an Amazon Redshift ARN ... 120
Example Policies for Amazon Redshift ... 121
Accessing Amazon Redshift Clusters and Databases ... 128
Using the Amazon Redshift Management Interfaces ... 128
Using the AWS SDK for Java ... 129
Signing an HTTP Request ... 131
Setting Up the Amazon Redshift CLI ... 134
Connecting to a Cluster ... 138

Configuring Connections in Amazon Redshift ... 138
Connecting to Clusters From Client Tools and Code ... 162
Troubleshooting Connection Issues in Amazon Redshift ... 173
Monitoring Cluster Performance ... 179
Overview ... 179
Summary of Performance Data ... 180
Amazon Redshift CloudWatch Metrics ... 180
Amazon Redshift Query/Load Performance Data ... 182
Working with Performance Data ... 183
Viewing Cluster Performance Data ... 184
Viewing Query Performance Data ... 186
Viewing Cluster Metrics During Load Operations ... 194
Creating an Alarm ... 195
Working with Performance Metrics in the Amazon CloudWatch Console ... 197
Events ... 199
Overview ... 199
Viewing Events Using the Console ... 199
Filtering Events ... 200
Viewing Events Using the AWS SDK for Java ... 201
View Events Using the Amazon Redshift CLI and API ... 202
Event Notifications ... 202
Overview ... 202
Amazon Redshift Event Categories and Event Messages ... 204
Managing Event Notifications Using the Amazon Redshift Console ... 209
Managing Event Notifications Using the Amazon Redshift CLI and API ... 214
Database Audit Logging ... 215
Overview ... 215
Amazon Redshift Logs ... 215
Connection Log ... 216
User Log ... 216
User Activity Log ... 217
Enabling Logging ... 217
Managing Log Files ... 218
Bucket Permissions for Amazon Redshift Audit Logging ... 218
Bucket Structure for Amazon Redshift Audit Logging ... 220
Troubleshooting Amazon Redshift Audit Logging ... 220
Using AWS CloudTrail for Amazon Redshift ... 221
Configuring Auditing Using the Console ... 221
Enabling Audit Logging Using the Console ... 221
Modifying the Bucket for Audit Logging ... 222
Disabling Audit Logging Using the Console ... 223
Configuring Logging by Using the Amazon Redshift CLI and API ... 223
Resizing Clusters ... 224
Overview ... 224
Resize Operation Overview ... 224
Snapshot, Restore, and Resize Operation Overview ... 225
Tutorial: Using the Resize Operation to Resize a Cluster ... 226
Prerequisites ... 227
Step 1: Resize the Cluster ... 227
Step 2: Delete the Sample Cluster ... 228
Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster ... 228
Prerequisites ... 229
Step 1: Take a Snapshot ... 229
Step 2: Restore the Snapshot into the Target Cluster ... 230
Step 3: Verify Data in the Target Cluster ... 231
Step 4: Resize the Target Cluster ... 232
Step 5: Copy Post-Snapshot Data from the Source to the Target Cluster ... 233
Step 6: Rename the Source and Target Clusters ... 234

Step 7: Delete the Source Cluster 235
Step 8: Clean Up Your Environment 236
Limits 237
Quotas and Limits 237
Naming Constraints 238
Tagging 240
Tagging Overview 240
Tagging Requirements 241
Managing Resource Tags Using the Console 241
How To Open the Manage Tags Window 242
How to Manage Tags in the Amazon Redshift Console 243
Managing Tags Using the Amazon Redshift API 243
Document History 245

What Is Amazon Redshift?

Welcome to the *Amazon Redshift Cluster Management Guide*. Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. You can start with just a few hundred gigabytes of data and scale to a petabyte or more. This enables you to use your data to acquire new insights for your business and customers.

The first step to create a data warehouse is to launch a set of nodes, called an Amazon Redshift cluster. After you provision your cluster, you can upload your data set and then perform data analysis queries. Regardless of the size of the data set, Amazon Redshift offers fast query performance using the same SQL-based tools and business intelligence applications that you use today.

Are You a First-Time Amazon Redshift User?

If you are a first-time user of Amazon Redshift, we recommend that you begin by reading the following sections:

- Amazon Redshift Management Overview (p. 2) – This topic provides an overview of Amazon Redshift.
- Service Highlights and Pricing – This product detail page provides the Amazon Redshift value proposition, service highlights, and pricing.
- Amazon Redshift Getting Started – This guide walks you through the process of creating a cluster, creating database tables, uploading data, and testing queries.
- Amazon Redshift Cluster Management Guide (*this guide*) – This guide shows you how to create and manage Amazon Redshift clusters.
- Amazon Redshift Database Developer Guide – If you are a database developer, this guide explains how to design, build, query, and maintain the databases that make up your data warehouse.

There are several ways to manage clusters. If you prefer a more interactive way of managing clusters, you can use the Amazon Redshift console or the AWS Command Line Interface (AWS CLI). If you are an application developer, you can use the Amazon Redshift Query API or the AWS Software Development Kit (SDK) libraries to manage clusters programmatically. If you use the Amazon Redshift Query API, you must authenticate every HTTP or HTTPS request to the API by signing it. For more information about signing requests, go to Signing an HTTP Request (p. 131).

For information about the CLI, API, and SDKs, go to the following links:

- AWS Command Line Interface Reference

- Amazon Redshift API Reference
- SDK References in Tools for Amazon Web Services

Amazon Redshift Management Overview

The Amazon Redshift service manages all of the work of setting up, operating, and scaling a data warehouse. These tasks include provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine.

Topics

- Cluster Management (p. 2)
- Cluster Access and Security (p. 2)
- Monitoring Clusters (p. 3)
- Databases (p. 4)

Cluster Management

An Amazon Redshift cluster is a set of nodes, which consists of a leader node and one or more compute nodes. The type and number of compute nodes that you need depends on the size of your data, the number of queries you will execute, and the query execution performance that you need.

Creating and Managing Clusters

Depending on your data warehousing needs, you can start with a small, single-node cluster and easily scale up to a larger, multi-node cluster as your requirements change. You can add or remove compute nodes to the cluster without any interruption to the service. For more information, see Amazon Redshift Clusters (p. 5).

Reserving Compute Nodes

If you intend to keep your cluster running for a year or longer, you can save money by reserving compute nodes for a one-year or three-year period. Reserving compute nodes offers significant savings compared to the hourly rates that you pay when you provision compute nodes on demand. For more information, see Purchasing Amazon Redshift Reserved Nodes (p. 107).

Creating Cluster Snapshots

Snapshots are point-in-time backups of a cluster. There are two types of snapshots: automated and manual. Amazon Redshift stores these snapshots internally in Amazon Simple Storage Service (Amazon S3) by using an encrypted Secure Sockets Layer (SSL) connection. If you need to restore from a snapshot, Amazon Redshift creates a new cluster and imports data from the snapshot that you specify. For more information about snapshots, see Amazon Redshift Snapshots (p. 82).

Cluster Access and Security

There are several features related to cluster access and security in Amazon Redshift. These features help you to control access to your cluster, define connectivity rules, and encrypt data and connections. These features are in addition to features related to database access and security in Amazon Redshift. For more information about database security, see Managing Database Security in the *Amazon Redshift Database Developer Guide*.

AWS Accounts and IAM Credentials

By default, an Amazon Redshift cluster is only accessible to the AWS account that creates the cluster. The cluster is locked down so that no one else has access. Within your AWS account, you use the AWS Identity and Access Management (IAM) service to create user accounts and manage permissions for those accounts to control cluster operations. For more information, see Controlling Access to Amazon Redshift Resources (p. 115).

Security Groups

By default, any cluster that you create is closed to everyone. IAM credentials only control access to the Amazon Redshift API-related resources: the Amazon Redshift console, command line interface (CLI), API, and SDK. To enable access to the cluster from SQL client tools via JDBC or ODBC, you use security groups:

- If you are using the EC2-Classic platform for your Amazon Redshift cluster, you must use Amazon Redshift security groups.
- If you are using the EC2-VPC platform for your Amazon Redshift cluster, you must use VPC security groups.

In either case, you add rules to the security group to grant explicit inbound access to a specific range of CIDR/IP addresses or to an Amazon Elastic Compute Cloud (Amazon EC2) security group if your SQL client runs on an Amazon EC2 instance. For more information, see Amazon Redshift Cluster Security Groups (p. 43).

In addition to the inbound access rules, you create database users to provide credentials to authenticate to the database within the cluster itself. For more information, see Databases (p. 4) in this topic.

Encryption

When you provision the cluster, you can optionally choose to encrypt the cluster for additional security. When you enable encryption, Amazon Redshift stores all data in user-created tables in an encrypted format. You can use either AWS Key Management Service (AWS KMS) or a hardware security module (HSM) to manage your Amazon Redshift encryption keys.

Encryption is an immutable property of the cluster. The only way to switch from an encrypted cluster to a nonencrypted cluster is to unload the data and reload it into a new cluster. Encryption applies to the cluster and any backups. When you restore a cluster from an encrypted snapshot, the new cluster is encrypted as well.

For more information about encryption, keys, and hardware security modules, see Amazon Redshift Database Encryption (p. 98).

SSL Connections

You can use Secure Sockets Layer (SSL) encryption to encrypt the connection between your SQL client and your cluster. For more information, see Configure Security Options for Connections (p. 158).

Monitoring Clusters

There are several features related to monitoring in Amazon Redshift. You can use database audit logging to generate activity logs, configure events and notification subscriptions to track information of interest, and use the metrics in Amazon Redshift and Amazon CloudWatch to learn about the health and performance of your clusters and databases.

Database Audit Logging

You can use the database audit logging feature to track information about authentication attempts, connections, disconnections, changes to database user definitions, and queries run in the database. This information is useful for security and troubleshooting purposes in Amazon Redshift. The logs are stored in Amazon S3 buckets. For more information, see Database Audit Logging (p. 215).

Events and Notifications

Amazon Redshift tracks events and retains information about them for a period of several weeks in your AWS account. For each event, Amazon Redshift reports information such as the date the event occurred, a description, the event source (for example, a cluster, a parameter group, or a snapshot), and the source ID. You can create Amazon Redshift event notification subscriptions that specify a set of event filters. When an event occurs that matches the filter criteria, Amazon Redshift uses Amazon Simple Notification Service to actively inform you that the event has occurred. For more information about events and notifications, see Amazon Redshift Events (p. 199).

Performance

Amazon Redshift provides performance metrics and data so that you can track the health and performance of your clusters and databases. Amazon Redshift uses Amazon CloudWatch metrics to monitor the physical aspects of the cluster, such as CPU utilization, latency, and throughput. Amazon Redshift also provides query and load performance data to help you monitor the database activity in your cluster. For more information about performance metrics and monitoring, see Monitoring Amazon Redshift Cluster Performance (p. 179).

Databases

Amazon Redshift creates one database when you provision a cluster. This is the database you use to load data and run queries on your data. You can create additional databases as needed by running a SQL command. For more information about creating additional databases, go to Step 1: Create a database in the *Amazon Redshift Database Developer Guide*.

When you provision a cluster, you specify a master user who has access to all of the databases that are created within the cluster. This master user is a superuser who is the only user with access to the database initially, though this user can create additional superusers and users. For more information, go to Superusers and Users in the *Amazon Redshift Database Developer Guide*.

Amazon Redshift uses parameter groups to define the behavior of all databases in a cluster, such as date presentation style and floating-point precision. If you don't specify a parameter group when you provision your cluster, Amazon Redshift associates a default parameter group with the cluster. For more information, see Amazon Redshift Parameter Groups (p. 56).

For more information about databases in Amazon Redshift, go to the Amazon Redshift Database Developer Guide.

Amazon Redshift Clusters

Topics

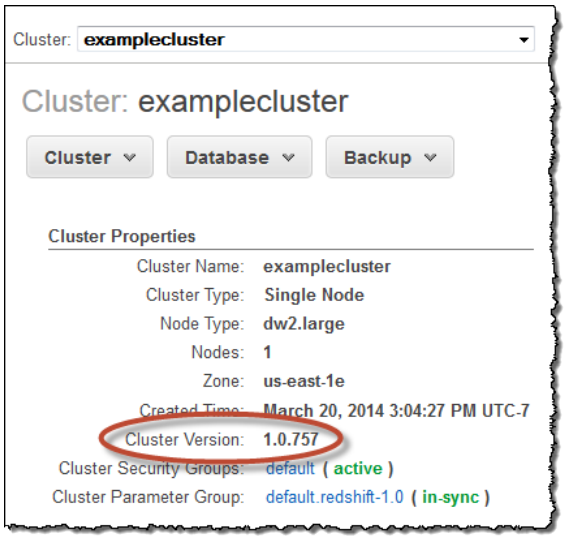
- Overview (p. 5)
- About Clusters and Nodes (p. 6)
- Resizing a Cluster (p. 8)
- Supported Platforms to Launch Your Cluster (p. 9)
- Regions and Availability Zone Considerations (p. 10)
- Maintenance Windows (p. 10)
- Default Disk Space Alarm (p. 11)
- Renaming Clusters (p. 11)
- Shutting Down and Deleting Clusters (p. 12)
- Managing Clusters Using the Console (p. 13)
- Managing Clusters Using the AWS SDK for Java (p. 29)
- Manage Clusters Using the Amazon Redshift CLI and API (p. 31)
- Managing Clusters in an Amazon Virtual Private Cloud (VPC) (p. 32)

Overview

An Amazon Redshift data warehouse is a collection of computing resources called *nodes*, which are organized into a group called a *cluster*. Each cluster runs an Amazon Redshift engine and contains one or more databases.

> **Note**
> At this time Amazon Redshift version 1.0 engine is available. However, as the engine is updated, multiple Amazon Redshift engine versions might be available for selection.

You can determine the Amazon Redshift engine and database versions for your cluster in the **Cluster Version** field in the console. The first two sections of the number are the cluster version, and the last section is the specific revision number of the database in the cluster. In the following example, the cluster version is 1.0 and the database revision number is 757.



Note
Although the console displays this information in one field, it is two parameters in the Amazon Redshift API: `ClusterVersion` and `ClusterRevisionNumber`. For more information, go to Cluster in the *Amazon Redshift API Reference*.

Amazon Redshift provides a setting, **Allow Version Upgrade**, to specify whether to automatically upgrade the Amazon Redshift engine in your cluster if a new version of the engine becomes available. This setting does not affect the database version upgrades, which are applied during the maintenance window that you specify for your cluster. Amazon Redshift engine upgrades are *major version upgrades*, and Amazon Redshift database upgrades are *minor version upgrades*. You can disable automatic version upgrades for major versions only. For more information about maintenance windows for minor version upgrades, see Maintenance Windows (p. 10).

About Clusters and Nodes

An Amazon Redshift cluster consists of nodes. There are two types of nodes: dense storage and dense compute. The *dense storage* node types are recommended for substantial data storage needs, while *dense compute* node types are optimized for performance-intensive workloads.

Each node type offers different sizes and limits to help you scale your cluster appropriately. The node size determines the storage capacity, memory, CPU, and price of each node in the cluster. The node limit determines how many nodes are available to the cluster. Initially, there is a limit of 16 nodes per account in each region; to have this limit increased, fill out the Node Limit Increase Form.

Some node sizes allow single-node clusters, while other node sizes require multi-node clusters. Each cluster has a *leader node* and a *compute node*. If you have a single-node cluster, the node is shared for leader and compute node functionality. If you have a multi-node cluster, the leader node is separate from the compute node or nodes.

A leader node receives queries from client applications, parses the queries and develops execution plans, which are an ordered set of steps to process these queries. The leader node then coordinates the parallel execution of these plans with the compute nodes, aggregates the intermediate results from these nodes and finally returns the results back to the client applications.

Compute nodes execute the steps specified in the execution plans and transmit data among themselves to serve these queries. The intermediate results are sent back to the leader node for aggregation before

being sent back to the client applications. For more information about leader nodes and compute nodes, go to Data warehouse system architecture in the *Amazon Redshift Database Developer Guide*.

The following tables outline the available node type combinations for Amazon Redshift. For more information and pricing, go to the Amazon Redshift pricing page.

Dense Storage Node Types

Node Size	Node Limits	Storage Capacity per Node	Maximum Storage Capacity per Cluster
dw1.xlarge	1 to 32	2 TB hard disk drive (HDD) storage	64 TB
dw1.8xlarge	2 to 128	16 TB hard disk drive (HDD) storage	2 PB

Note
The *dw1.xlarge node size* was previously called *dw.hs1.xlarge* and the *dw1.8xlarge node size* was previously called *dw.hs1.8xlarge*. You can use either value in the Amazon Redshift API and CLI, though we recommend that you update any scripts that reference the old names to use the new names instead. The Amazon Redshift console uses the new names.

Dense Compute Node Types

Node Size	Node Limits	Storage Capacity per Node	Maximum Storage Capacity per Cluster
dw2.large	1 to 32	160 GB solid state drive (SSD) storage	5.12 TB
dw2.8xlarge	2 to 128	2.56 TB solid state drive (SSD) storage	326 TB

Determining the Number of Nodes

The number of nodes that you choose depends on the size of your data set and your desired query performance. Using the dense storage node type as an example, if you have 32 TB of data, you can choose sixteen dw1.xlarge nodes or two dw1.8xlarge nodes. If your data grows in small increments, choosing the dw1.xlarge node size will allow you to scale in increments of 2 TB. If you typically see data growth in larger increments, a dw1.8xlarge node size might be a better choice.

Because Amazon Redshift distributes and executes queries in parallel across all of a cluster's compute nodes, you can increase query performance by adding nodes to your cluster. Amazon Redshift also distributes your data across all compute nodes in a cluster. When you run a cluster with at least two compute nodes, data on each node will always be mirrored on disks on another node and you reduce the risk of incurring data loss.

Regardless of the choice you make, you can monitor query performance in the Amazon Redshift console and with Amazon CloudWatch metrics. You can also add or remove nodes as needed to achieve the balance between storage and performance that works best for you. When you request an additional node, Amazon Redshift takes care of all the details of deployment, load balancing, and data maintenance. For more information about cluster performance, see Monitoring Amazon Redshift Cluster Performance (p. 179).

If you intend to keep your cluster running continuously for a prolonged period, say, one year or more, you can pay considerably less by reserving the compute nodes for a one-year or three-year period. To reserve

compute nodes, you purchase what are called reserved node offerings. You purchase one offering for each compute node that you want to reserve. When you reserve a compute node, you pay a fixed up-front charge and then an hourly recurring charge, whether your cluster is running or not. The hourly charges, however, are significantly lower than those for on-demand usage. For more information, see Purchasing Amazon Redshift Reserved Nodes (p. 107).

Resizing a Cluster

If your storage and performance needs change after you initially provision your cluster, you can resize your cluster. You can scale the cluster in or out by adding or removing nodes. Additionally, you can scale the cluster up or down by specifying a different node type.

For example, you can add more nodes, change node types, change a single-node cluster to a multi-node cluster, or change a multi-node cluster to a single-node cluster. You must, however, ensure that the resulting cluster is large enough to hold the data that you currently have or else the resize will fail. When using the API, you have to specify both the node type and the number of nodes even if you only change one of the two.

When you resize a cluster, Amazon Redshift first puts your existing cluster into read-only mode. Amazon Redshift then provisions the nodes as requested, copies data to the new cluster, and switches your connections to use the new cluster. At this time, you lose any open connections to the old cluster. If you have any queries in progress at the time of the switch, you will notice the connection loss. You must restart the query on the new cluster. You should wait until the resize operation is complete before you resume loading data or performing queries on the cluster.

When you resize your cluster, it will remain in read-only mode until the resize finishes. You can view the resize progress on the cluster's **Status** tab in the Amazon Redshift console. The time it takes to resize a cluster depends on the amount of data in each node. Typically, the resize process varies from a couple of hours to a day, although clusters with larger amounts of data might take even longer. This is because the data is copied in parallel from each node on the source cluster to the nodes in the target cluster. For more information about resizing clusters, see Tutorial: Resizing Clusters in Amazon Redshift (p. 224) and Resizing a Cluster (p. 24).

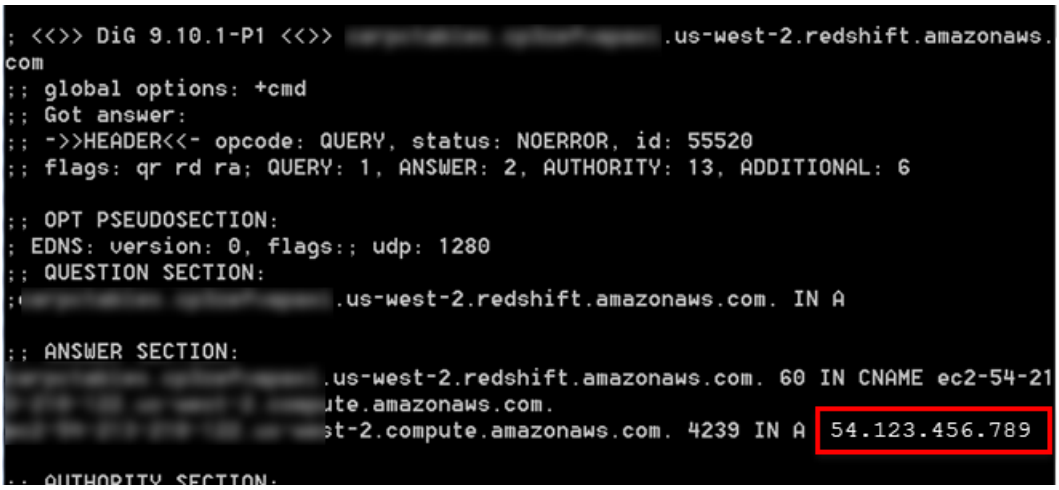
Amazon Redshift does not sort tables during a resize operation. When you resize a cluster, Amazon Redshift distributes the database tables to the new compute nodes based on their distribution styles and runs an ANALYZE to update statistics. Rows that are marked for deletion are not transferred, so you will only need to run a VACUUM if your tables need to be resorted. For more information, see Vacuuming tables in the *Amazon Redshift Database Developer Guide*.

If your cluster is public and is in a VPC, it keeps the same elastic IP address (EIP) for the leader node after resizing. If your cluster is private and is in a VPC, it keeps the same private IP address for the leader node after resizing. If your cluster is not in a VPC, a new public IP address is assigned for the leader node as part of the resize operation.

To get the leader node IP address for a cluster, use the dig utility, as shown following:

```
dig mycluster.abcd1234.us-west-2.redshift.amazonaws.com
```

The leader node IP address is at the end of the ANSWER SECTION in the results, as shown following:


; <<>> DiG 9.10.1-P1 <<>> .us-west-2.redshift.amazonaws.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55520
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 13, ADDITIONAL: 6
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
.us-west-2.redshift.amazonaws.com. IN A
;; ANSWER SECTION:
.us-west-2.redshift.amazonaws.com. 60 IN CNAME ec2-54-21
ute.amazonaws.com.
st-2.compute.amazonaws.com. 4239 IN A 54.123.456.789
;; AUTHORITY SECTION:

You can get the dig utility as part of the BIND software download. For more information on BIND, go to BIND in the Internet Systems Consortium documentation.

Supported Platforms to Launch Your Cluster

Amazon Redshift clusters run in Amazon Elastic Compute Cloud (Amazon EC2) instances that are configured for the Amazon Redshift node type and size that you select. You can launch an Amazon Redshift cluster in one of two platforms: EC2-Classic or EC2-VPC, which are the supported platforms for Amazon EC2 instances. For more information about these platforms, go to Supported Platforms in the *Amazon EC2 User Guide for Linux Instances*. The platform or platforms available to you depend on your AWS account settings.

> **Note**
> To prevent connection issues between SQL client tools and the Amazon Redshift database, we recommend disabling TCP/IP jumbo frames by setting the maximum transmission unit (MTU) to 1500 on the network interface (NIC) of your Amazon EC2 instances. For more information about setting the MTU, see Queries Appear to Hang in the Client and Do Not Reach the Cluster (p. 176).

EC2-Classic Platform

In the EC2-Classic platform, your cluster runs in a single, flat network that you share with other AWS customers. If you provision your cluster in the EC2-Classic platform, you control access to your cluster by associating one or more Amazon Redshift cluster security groups with the cluster. For more information, see Amazon Redshift Cluster Security Groups (p. 43).

EC2-VPC Platform

In the EC2-VPC platform, your cluster runs in a virtual private cloud (VPC) that is logically isolated to your AWS account. If provision your cluster in the EC2-VPC platform, you control access to your cluster by associating one or more VPC security groups with the cluster. For more information, go to Security Groups for Your VPC in the *Amazon VPC User Guide*.

To create a cluster in a VPC, you must first create an Amazon Redshift cluster subnet group by providing subnet information of your VPC, and then provide the subnet group when launching the cluster. For more information, see Amazon Redshift Cluster Subnet Groups (p. 36).

For more information about Amazon Virtual Private Cloud (Amazon VPC), go to the Amazon VPC product detail page.

Choose a Platform

Your AWS account is capable of launching instances either into both platforms, or only into EC2-VPC, on a region-by-region basis. To determine which platform your account supports, and then launch a cluster, do the following:

1. Decide on the AWS region in which you want to deploy a cluster. For a list of AWS regions in which Amazon Redshift is available, go to Regions and Endpoints in the *Amazon Web Services General Reference*.
2. Find out which Amazon EC2 platforms your account supports in the chosen AWS region. You can find this information in the Amazon EC2 console. For step-by-step instructions, go to Supported Platforms in the *Amazon EC2 User Guide for Linux Instances*.
3. If your account supports both of the platforms, choose the one on which you want to deploy your Amazon Redshift cluster. If your account supports only EC2-VPC, you must deploy your cluster in VPC.
4. Deploy your Amazon Redshift cluster. You can deploy a cluster by using the Amazon Redshift console, or programmatically by using the Amazon Redshift API, CLI, or SDK libraries. For more information about these options and links to the related documentation, see What Is Amazon Redshift? (p. 1).

Regions and Availability Zone Considerations

Amazon Redshift is available in several AWS regions. By default, Amazon Redshift provisions your cluster in a randomly selected Availability Zone (AZ) within the AWS region that you select. All the cluster nodes are provisioned in the same AZ.

You can optionally request a specific AZ if Amazon Redshift is available in that AZ. For example, if you already have an Amazon EC2 instance running in one AZ, you might want to create your Amazon Redshift cluster in the same AZ to reduce latency. On the other hand, you might want to choose another AZ for higher availability. Amazon Redshift might not be available in all AZs within a region.

For a list of supported AWS regions where you can provision an Amazon Redshift cluster, go to Regions and Endpoints in the *Amazon Web Services General Reference*.

Maintenance Windows

Amazon Redshift periodically performs maintenance to apply upgrades to your cluster. During these updates, your Amazon Redshift cluster is not available for normal operations.

Amazon Redshift assigns a 30 minute maintenance window at random from an 8-hour block of time per region, occurring on a random day of the week (Monday through Sunday, inclusive). The following list shows the time blocks for each region from which the default maintenance windows are assigned:

- US East (N. Virginia) region: 03:00-11:00 UTC
- US West (Oregon) region: 06:00-14:00 UTC
- EU (Frankfurt) region: 06:00-14:00 UTC
- Asia Pacific (Sydney) region: 12:00-20:00 UTC
- Asia Pacific (Singapore) region: 14:00-22:00 UTC
- Asia Pacific (Tokyo) region: 17:00-03:00 UTC

- EU (Ireland) region: 22:00-06:00 UTC

While Amazon Redshift is performing maintenance, it terminates any queries or other operations that are in progress. If there are no maintenance tasks to perform during the scheduled maintenance window, your cluster continues to operate normally until the next scheduled maintenance window.

You can change the scheduled maintenance window by modifying the cluster, either programmatically or by using the Amazon Redshift console. The window must be at least 30 minutes and not longer than 24 hours. For more information, see Managing Clusters Using the Console (p. 13).

Default Disk Space Alarm

When you create an Amazon Redshift cluster, you can optionally configure an Amazon CloudWatch alarm to monitor the average percentage of disk space that is used across all of the nodes in your cluster. We'll refer to this alarm as the *default disk space alarm*.

The purpose of default disk space alarm is to help you monitor the storage capacity of your cluster. You can configure this alarm based on the needs of your data warehouse. For example, you can use the warning as an indicator that you might need to resize your cluster, either to a different node type or to add nodes, or perhaps to purchase reserved nodes for future expansion.

The default disk space alarm triggers when disk usage reaches or exceeds a specified percentage for a certain number of times and at a specified duration. By default, this alarm triggers when the percentage that you specify is reached, and stays at or above that percentage for five minutes or longer. You can edit the default values after you launch the cluster.

When the CloudWatch alarm triggers, Amazon Simple Notification Service (Amazon SNS) sends a notification to specified recipients to warn them that the percentage threshold is reached. Amazon SNS uses a topic to specify the recipients and message that are sent in a notification. You can use an existing Amazon SNS topic; otherwise, a topic is created based on the settings that you specify when you launch the cluster. You can edit the topic for this alarm after you launch the cluster. For more information about creating Amazon SNS topics, see Getting Started with Amazon Simple Notification Service.

After you launch the cluster, you can view and edit the alarm from the cluster's **Status** window under **CloudWatch Alarms**. The name is **percentage-disk-space-used-default-<*string*>**. You can open the alarm to view the Amazon SNS topic that it is associated with and edit alarm settings. If you did not select an existing Amazon SNS topic to use, the one created for you is named **<*clustername*>-default-alarms (<*recipient*>)**; for example, **examplecluster-default-alarms (notify@example.com)**.

For more information about configuring and editing the default disk space alarm, see Creating a Cluster (p. 14) and Editing the Default Disk Space Alarm (p. 28).

> **Note**
> If you delete your cluster, the alarm associated with the cluster will not be deleted but it will not trigger. You can delete the alarm from the CloudWatch console if you no longer need it.

Renaming Clusters

You can rename a cluster if you want the cluster to use a different name. Because the endpoint to your cluster includes the cluster name (also referred to as the *cluster identifier*), the endpoint will change to use the new name after the rename finishes. For example, if you have a cluster named `examplecluster` and rename it to `newcluster`, the endpoint will change to use the `newcluster` identifier. Any applications that connect to the cluster must be updated with the new endpoint.

You might rename a cluster if you want to change the cluster to which your applications connect without having to change the endpoint in those applications. In this case, you must first rename the original cluster and then change the second cluster to reuse the name of the original cluster prior to the rename. Doing this is necessary because the cluster identifier must be unique within your account and region, so the original cluster and second cluster cannot have the same name . You might do this if you restore a cluster from a snapshot and don't want to change the connection properties of any dependent applications.

Note
If you delete the original cluster, you are responsible for deleting any unwanted cluster snapshots.

When you rename a cluster, the cluster status changes to `renaming` until the process finishes. The old DNS name that was used by the cluster is immediately deleted, although it could remain cached for a few minutes. The new DNS name for the renamed cluster becomes effective within about 10 minutes. The renamed cluster is not available until the new name becomes effective. The cluster will be rebooted and any existing connections to the cluster will be dropped. After this completes, the endpoint will change to use the new name. For this reason, you should stop queries from running before you start the rename and restart them after the rename finishes.

Cluster snapshots are retained, and all snapshots associated with a cluster remain associated with that cluster after it is renamed. For example, suppose you have a cluster that serves your production database and the cluster has several snapshots. If you rename the cluster and then replace it in the production environment with a snapshot, the cluster that you renamed will still have those existing snapshots associated with it.

Amazon CloudWatch alarms and Amazon Simple Notification Service (Amazon SNS) event notifications are associated with the name of the cluster. If you rename the cluster, you need to update these accordingly. You can update the CloudWatch alarms in the CloudWatch console, and you can update the Amazon SNS event notifications in the Amazon Redshift console on the **Events** pane. The load and query data for the cluster continues to display data from before the rename and after the rename. However, performance data is reset after the rename process finishes.

For more information, see Modifying a Cluster (p. 20).

Shutting Down and Deleting Clusters

You can shut down your cluster if you want to stop it from running and incurring charges. When you shut it down, you can optionally create a final snapshot. If you create a final snapshot, Amazon Redshift will create a manual snapshot of your cluster before shutting it down. You can later restore that snapshot if you want to resume running the cluster and querying data.

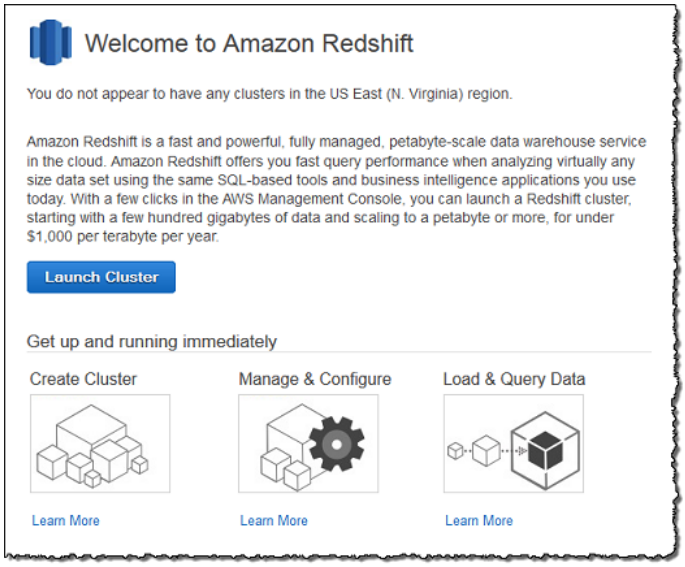
If you no longer need your cluster and its data, you can shut it down without creating a final snapshot. In this case, the cluster and data are deleted permanently. For more information about shutting down and deleting clusters, see Shutting Down or Deleting a Cluster (p. 22).

Regardless of whether you shut down your cluster with a final manual snapshot, all automated snapshots associated with the cluster will be deleted after the cluster is shut down. Any manual snapshots associated with the cluster are retained. Any manual snapshots that are retained, including the optional final snapshot, are charged at the Amazon Simple Storage Service storage rate if you have no other clusters running when you shut down the cluster, or if you exceed the available free storage that is provided for your running Amazon Redshift clusters. For more information about snapshot storage charges, go to the Amazon Redshift pricing page.

Managing Clusters Using the Console

This topic explains how to work with clusters in the Amazon Redshift console. You can use the **Clusters** section in the Amazon Redshift console to create, modify, resize, delete, reboot, and back up clusters.

When you don't have any clusters in a region, and you navigate to the **Clusters** page, you will see an option to launch a cluster. In the following screenshot, the region is the US East (N. Virginia) region and there are no clusters for this account.


Welcome to Amazon Redshift
You do not appear to have any clusters in the US East (N. Virginia) region.
Amazon Redshift is a fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud. Amazon Redshift offers you fast query performance when analyzing virtually any size data set using the same SQL-based tools and business intelligence applications you use today. With a few clicks in the AWS Management Console, you can launch a Redshift cluster, starting with a few hundred gigabytes of data and scaling to a petabyte or more, for under $1,000 per terabyte per year.
Launch Cluster
Get up and running immediately
Create Cluster
Manage & Configure
Load & Query Data
Learn More
Learn More
Learn More

When you have at least one cluster in the region that you have selected, the **Clusters** section displays a subset of information about all the clusters for the account in that region. In the following screenshot, there is one cluster created for this account in the selected region.

Topics

- Creating a Cluster (p. 14)
- Modifying a Cluster (p. 20)
- Shutting Down or Deleting a Cluster (p. 22)
- Rebooting a Cluster (p. 23)
- Resizing a Cluster (p. 24)
- Getting Information About Cluster Configuration (p. 26)
- Getting an Overview of Cluster Status (p. 26)
- Taking a Snapshot of a Cluster (p. 27)
- Editing the Default Disk Space Alarm (p. 28)
- Working with Cluster Performance Data (p. 29)

Creating a Cluster

Before you create a cluster, review the information in the Overview (p. 5) of this section.

To create a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. Click **Launch Cluster**.
3. On the **Cluster Details** page, specify values for the following options, and then click **Continue**.

   Cluster Identifier
   Type a unique name for your cluster.

   Cluster identifiers must meet the following conditions:

   - They must contain from 1 to 63 alphanumeric characters or hyphens.
   - Alphabetic characters must be lowercase.
   - The first character must be a letter.
   - They cannot end with a hyphen or contain two consecutive hyphens.
   - They must be unique for all clusters within an AWS account.

   Database Name
   Type a name if you want to create a database with a custom name (for example, mydb). This field is optional. A default database named *dev* is created for the cluster whether or not you specify a custom database name.

   Database names must meet the following conditions:

   - They must contain 1 to 64 alphanumeric characters.
   - They must contain only lowercase letters.
   - A database name cannot be a reserved word. For more information, go to Reserved Words in the *Amazon Redshift Database Developer Guide*.

   Database Port
   Type a port number through which you will connect from client applications to the database. The port number must be included in the connection string when opening JDBC or ODBC connections to the databases in the cluster.

   The port number must meet the following conditions:

   - It must contain only numeric characters.
   - It must fall in the range of 1150 to 65535. The default port is 5439.
   - It must specify an open port that accepts inbound connections, if you are behind a firewall.

   Master User Name
   Type an account name for the master user of the database.

   Master user names must meet the following conditions:

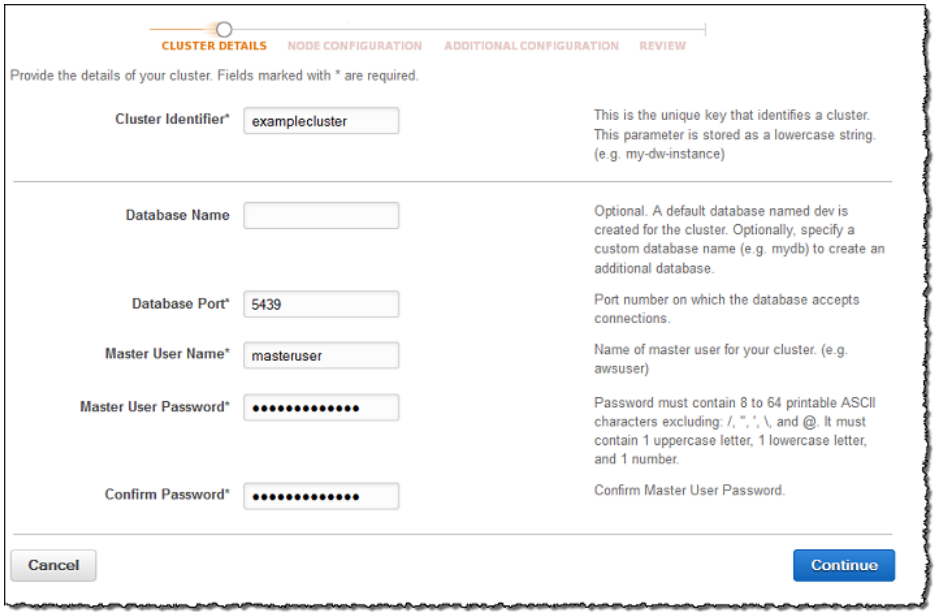
   - They must contain from 1 to 128 alphanumeric characters.
   - The first character must be a letter.
   - A master user name cannot be a reserved word. For more information, go to Reserved Words in the *Amazon Redshift Database Developer Guide*.

   Master User Password and **Confirm Password**
   Type a password for the master user account, and then retype it to confirm the password.

   The password must meet the following conditions:

- It must be from 8 to 64 characters in length.
- It must contain at least one uppercase letter.
- It must contain at least one lowercase letter.
- It must contain at least one number.
- It can be any printable ASCII character (ASCII code 33 to 126) except single quotation mark, double quotation mark, \, /, @, or space.

In the following screenshot, `examplecluster` is the cluster identifier, no custom database name is specified, 5439 is the port, and `masteruser` is the master user name.


CLUSTER DETAILS
NODE CONFIGURATION
ADDITIONAL CONFIGURATION
REVIEW
Provide the details of your cluster. Fields marked with * are required.
Cluster Identifier*
examplecluster
This is the unique key that identifies a cluster. This parameter is stored as a lowercase string. (e.g. my-dw-instance)
Database Name
Optional. A default database named dev is created for the cluster. Optionally, specify a custom database name (e.g. mydb) to create an additional database.
Database Port*
5439
Port number on which the database accepts connections.
Master User Name*
masteruser
Name of master user for your cluster. (e.g. awsuser)
Master User Password*
Password must contain 8 to 64 printable ASCII characters excluding: /, ", ', \, and @. It must contain 1 uppercase letter, 1 lowercase letter, and 1 number.
Confirm Password*
Confirm Master User Password.
Cancel
Continue

4. On the **Node Configuration** page, specify values for the following options, and then click **Continue**.

   Node Type

   Select a node type. When you select a node type, the page displays information that corresponds to the selected node type, such as **CPU**, **Memory**, **Storage**, and **I/O Performance**.

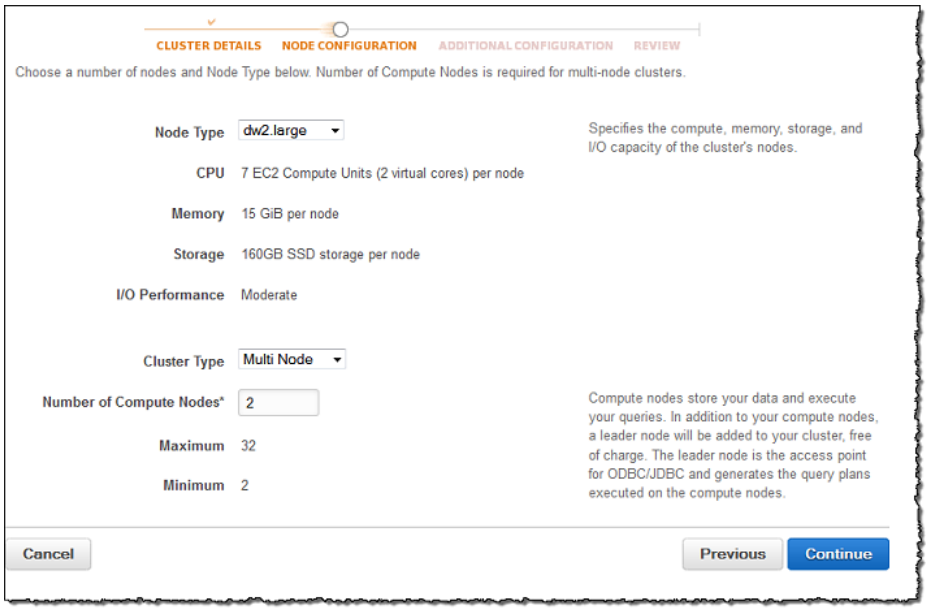
   Cluster Type

   Select a cluster type. When you do, the maximum number of compute nodes for the selected node and cluster type appears in the **Maximum** box, and the minimum number appears in the **Minimum** box.

   If you click **Single Node**, you will have one node that shares leader and compute functionality.

   If you click **Multi Node**, specify the number of compute nodes that you want for the cluster in **Number of Compute Nodes**.

   In the following screenshot, the **dw2.large** node type is selected for a **Multi Node** cluster with two compute nodes.


CLUSTER DETAILS
NODE CONFIGURATION
ADDITIONAL CONFIGURATION
REVIEW
Choose a number of nodes and Node Type below. Number of Compute Nodes is required for multi-node clusters.
Node Type
dw2.large
Specifies the compute, memory, storage, and I/O capacity of the cluster's nodes.
CPU
7 EC2 Compute Units (2 virtual cores) per node
Memory
15 GiB per node
Storage
160GB SSD storage per node
I/O Performance
Moderate
Cluster Type
Multi Node
Number of Compute Nodes*
2
Maximum
32
Minimum
2
Compute nodes store your data and execute your queries. In addition to your compute nodes, a leader node will be added to your cluster, free of charge. The leader node is the access point for ODBC/JDBC and generates the query plans executed on the compute nodes.
Cancel
Previous
Continue

5. On the **Additional Configuration** page, specify values for the following options, and then click **Continue**.

   a. Under **Provide the optional additional configuration details below**, configure the following options:

      Cluster Parameter Group
      Select a cluster parameter group to associate with the cluster. If you don't select one, the cluster uses the default parameter group.

      Encrypt Database
      Select whether you want to encrypt all data within the cluster and its snapshots. If you leave the default setting, **None**, encryption is not enabled. If you want to enable encryption, select whether you want to use AWS Key Management Service (AWS KMS) or a hardware security module (HSM), and then configure the related settings. For more information about encryption in Amazon Redshift, see Amazon Redshift Database Encryption (p. 98).

      - **KMS**

        Click **KMS** if you want to enable encryption and use AWS KMS to manage your encryption key.

        Note
        AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud. You can access AWS KMS from the **Encryption Keys** section of the AWS Identity and Access Management console or the AWS KMS APIs to centrally create encryption keys, define the policies that control how keys can be used, and audit key usage to prove they are being used correctly. For more information about managing your Amazon Redshift encryption key using AWS KMS, go to AWS Key Management Service Developer Guide.

        After you click **KMS**, you can select a key from the **Master Key** list.

        If you select **(default) aws/redshift**, Amazon Redshift will use a default customer master key (CMK). The first time you create an encrypted Amazon Redshift cluster in a region, a default CMK is created for you automatically. This key is used for Amazon Redshift encryption unless you select a CMK that you created separately using AWS KMS. Creating

your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls for, and audit the encryption keys used to protect your data.

If you want to use another CMK from your account, you need to create it first in AWS KMS. Then the key will be available in the **Master Key** list in Amazon Redshift. When you select a CMK, the Amazon Redshift console updates with information in **Description**, **Account**, and **KMS Key ID** so you can verify the details of the selected key. For more information about creating CMKs, go to Creating Customer Master Keys in the *AWS Key Management Service Developer Guide*.

If you want to use a key from another account, select **Enter a key ARN** from **Master Key**. Then type the ARN for the key to use. You must have permission to use the key. For more information about access to keys in AWS KMS, go to Controlling Access to Your Keys in the *AWS Key Management Service Developer Guide*.

- **HSM**

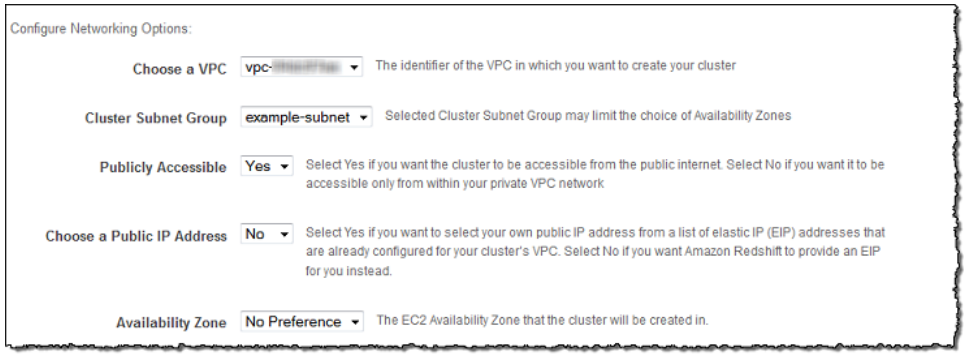
  Click **HSM** if you want to enable encryption and use a hardware security module (HSM) to manage your encryption key.

  If you click **HSM**, select values from **HSM Connection** and **HSM Client Certificate**. These values are required for Amazon Redshift and the HSM to form a trusted connection over which the cluster key can be passed. The HSM connection and client certificate must be set up in Amazon Redshift before you launch a cluster. For more information about setting up HSM connections and client certificates, see Hardware Security Modules (p. 99).

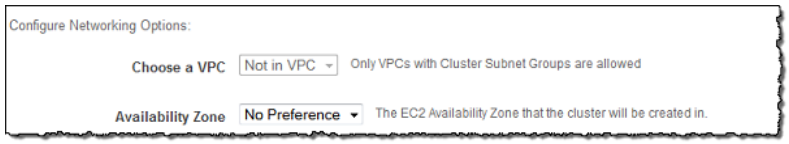
b. Under **Configure Networking Options**, you configure whether to launch your cluster in a virtual private cloud (VPC) or outside a VPC. The option you choose affects the additional options available in this section. Amazon Redshift uses the EC2-Classic and EC2-VPC platforms to launch clusters. Your AWS account determines which platform or platforms are available to you for your cluster. For more information, see Supported Platforms in the *Amazon EC2 User Guide for Linux Instances*.

Choose a VPC

If you want to launch your cluster in a virtual private cloud (VPC), select the VPC you want to use. You must have at least one Amazon Redshift subnet group set up to use VPCs. For more information, see Amazon Redshift Cluster Subnet Groups (p. 36).


Configure Networking Options:
Choose a VPC
vpc-
The identifier of the VPC in which you want to create your cluster
Cluster Subnet Group
example-subnet
Selected Cluster Subnet Group may limit the choice of Availability Zones
Publicly Accessible
Yes
Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network
Choose a Public IP Address
No
Select Yes if you want to select your own public IP address from a list of elastic IP (EIP) addresses that are already configured for your cluster's VPC. Select No if you want Amazon Redshift to provide an EIP for you instead.
Availability Zone
No Preference
The EC2 Availability Zone that the cluster will be created in.

If you want to launch your cluster outside a VPC, click **Not in VPC**. This option is available only to AWS accounts that support the EC2-Classic platform. Otherwise, you must launch your cluster in a VPC.


Configure Networking Options:
Choose a VPC
Not in VPC
Only VPCs with Cluster Subnet Groups are allowed
Availability Zone
No Preference
The EC2 Availability Zone that the cluster will be created in.

Cluster Subnet Group
Select the Amazon Redshift subnet group in which to launch the cluster.

> **Note**
> This option is available only for clusters in a VPC.

Publicly Accessible
Click **Yes** to enable connections to the cluster from outside of the VPC in which you launch the cluster. Click **No** if you want to limit connections to the cluster from only within the VPC.

> **Note**
> This option is available only for clusters in a VPC.

Choose a Public IP Address
If you set **Publicly Accessible** to **Yes**, click **No** here to have Amazon Redshift to provide an Elastic IP (EIP) for the cluster, or click **Yes** if you want to use an EIP that you have created and manage. If you have Amazon Redshift create the EIP, it is managed by Amazon Redshift.

> **Note**
> This option is available only for clusters in a VPC where **Publicly Accessible** is **Yes**.

Elastic IP
Select the EIP that you want to use to connect to the cluster from outside of the VPC.

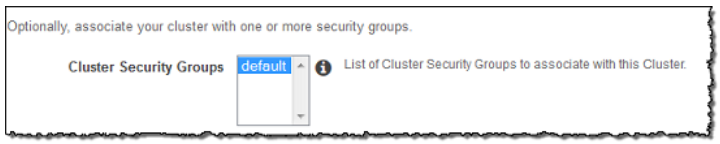
> **Note**
> This option is available only for clusters in a VPC where **Publicly Accessible** and **Choose a Public IP Address** are **Yes**.

Availability Zone
Click **No Preference** to have Amazon Redshift select the Availability Zone that the cluster will be created in. Otherwise, select a specific Availability Zone.

c. Under **Optionally, associate your cluster with one or more security groups**, specify values for the following options:

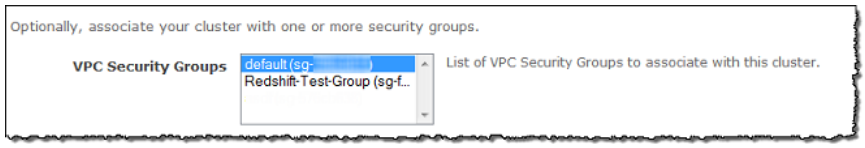
Cluster Security Groups
Select an Amazon Redshift security group or groups for the cluster. By default, the selected security group is the default security group. For more information about cluster security groups, see Amazon Redshift Cluster Security Groups (p. 43).

> **Note**
> This option is only available if you launch your cluster in the EC2-Classic platform.


Optionally, associate your cluster with one or more security groups.
Cluster Security Groups
default
List of Cluster Security Groups to associate with this Cluster.

VPC Security Groups
Select a VPC security group or groups for the cluster. By default, the selected security group is the default VPC security group. For more information about VPC security groups, go to Security Groups for Your VPC in the *Amazon VPC User Guide*.

> **Note**
> This option is only available if you launch your cluster in the EC2-VPC platform.

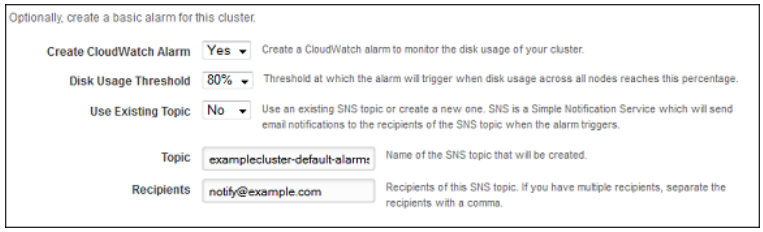

Optionally, associate your cluster with one or more security groups.
VPC Security Groups
default (sg-
Redshift-Test-Group (sg-f...
List of VPC Security Groups to associate with this cluster.

d. Under **Optionally create a basic alarm for this cluster**, configure the following options, and then click **Continue**:

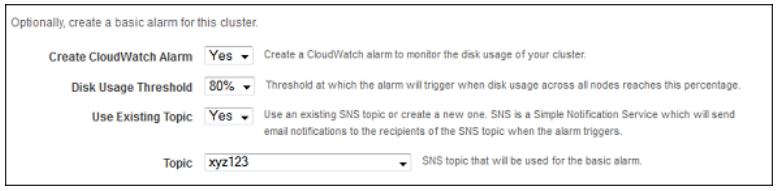
Create CloudWatch Alarm
Click **Yes** if you want to create an alarm that monitors the disk usage of your cluster, and then specify values for the corresponding options. Click **No** if you don't want to create an alarm.

Disk Usage Threshold
Select a percentage of average disk usage that has been reached or exceeded at which the alarm should trigger.

Use Existing Topic
Click **No** if you want to create a new Amazon SNS topic for this alarm. In the **Topic** box, edit the default name if necessary. In the **Recipients** box, type the email addresses for any recipients who should receive the notification when the alarm triggers.


Optionally, create a basic alarm for this cluster.
Create CloudWatch Alarm
Yes
Create a CloudWatch alarm to monitor the disk usage of your cluster.
Disk Usage Threshold
80%
Threshold at which the alarm will trigger when disk usage across all nodes reaches this percentage.
Use Existing Topic
No
Use an existing SNS topic or create a new one. SNS is a Simple Notification Service which will send email notifications to the recipients of the SNS topic when the alarm triggers.
Topic
examplecluster-default-alarm
Name of the SNS topic that will be created.
Recipients
notify@example.com
Recipients of this SNS topic. If you have multiple recipients, separate the recipients with a comma.

Click **Yes** if you want to select an existing Amazon SNS topic for this alarm, and then in the **Topic** list, select the topic that you want to use.


Optionally, create a basic alarm for this cluster.
Create CloudWatch Alarm
Yes
Create a CloudWatch alarm to monitor the disk usage of your cluster.
Disk Usage Threshold
80%
Threshold at which the alarm will trigger when disk usage across all nodes reaches this percentage.
Use Existing Topic
Yes
Use an existing SNS topic or create a new one. SNS is a Simple Notification Service which will send email notifications to the recipients of the SNS topic when the alarm triggers.
Topic
xyz123
SNS topic that will be used for the basic alarm.

6. On the **Review** page, review the details of the cluster. If everything is satisfactory, click **Launch Cluster** to start the creation process. Otherwise, click **Back** to make any necessary changes, and then click **Continue** to return to the **Review** page.

   Note
   Some cluster properties, such as the values for **Database Port** and **Master User Name**, cannot be modified later. If you need to change them, click **Back** to change them now.

   The following screenshot shows a summary of various options selected during the cluster launch process.

CLUSTER DETAILS
NODE CONFIGURATION
ADDITIONAL CONFIGURATION
REVIEW
You are about to launch a cluster with following the following specifications:
Cluster Properties
These attributes specify the name of your cluster, what type of virtual hardware it will run on, how many nodes it will contain, and the availability zone in which it will be located.
Cluster Identifier: examplecluster
Node Type: dw2.large
Number of Compute Nodes: 2 (plus a free leader node)
Availability Zone: No Preference
Database Configuration
These properties specify the database name, port, and username you will use to connect to the database. The parameter group contains configuration values used by the database.
Database Name: A default database will be created (dev)
Database Port: 5439
Master User Name: masteruser
Cluster Parameter Group: default.redshift-1.0
Security, Access, and Encryption
These settings control whether your cluster will be created in an existing VPC to allow for simpler integration with other AWS Services, and the security groups which define access rules to your cluster.
Virtual Private Cloud: vpc-
Cluster Subnet Group: example-subnet
Publicly Accessible: Yes
Elastic IP: Not used
VPC Security Groups: sg-
Encrypt Database: Yes
Use HSM: Yes
HSM Connection: hsm-example
HSM Client Certificate: cert-example
CloudWatch Alarms
CloudWatch alarms are used to notify if metrics for your cluster are within a certain threshold. All recipients under the SNS topic specified for your alarm will receive notifications once an alarm is triggered.
Topic: examplecluster-default-alarms
Recipients: notify@examplecluster.com
Threshold: 80%

7. After you initiate the creation process, click **Close**. The cluster might take several minutes to be ready to use.

Cluster examplecluster is being created.
Note: Your cluster may take a few minutes to launch.
View your cluster on the Clusters dashboard.
Close

   You can monitor the status of the operation in the performance dashboard.

Modifying a Cluster

You can modify the following configuration options for an existing cluster:

- **Cluster Identifier**
- **Cluster Parameter Group**
- **VPC Security Groups**
- **Master User Password**
- **HSM Connection**
- **HSM Client Certificate**
- **Automated Snapshot Retention Period**
- **Maintenance Window Start**
- **Maintenance Window End**

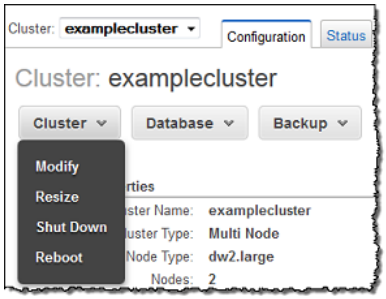
All of these changes take effect only after the cluster is restarted.

If you decrease the automated snapshot retention period, existing automated snapshots whose settings fall outside of the new retention period are deleted. For information about working with snapshots, see Amazon Redshift Snapshots (p. 82).

To modify a cluster

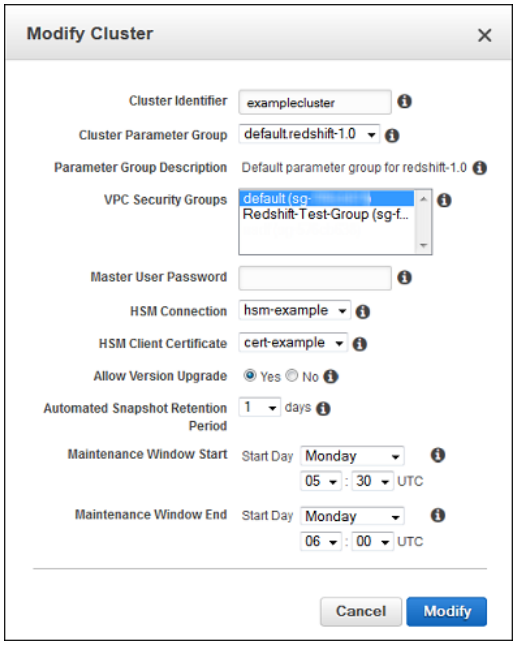
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster that you want to modify.

3. On the **Configuration** tab of the cluster details page, click **Cluster**, and then click **Modify**.



4. In the **Modify Cluster** window, make the changes to the cluster, and then click **Modify**.

   The following screenshot shows the **Modify Cluster** options.



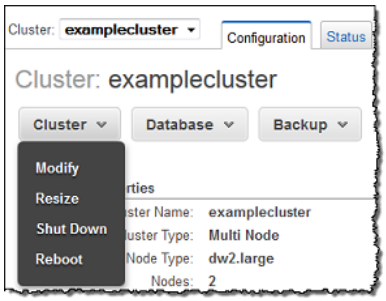
Shutting Down or Deleting a Cluster

If you no longer need your cluster, you can shut it down or delete it. If you plan to provision a new cluster with the same data and configuration as the one you are shutting down, you will need a snapshot so that you can restore the snapshot at a later time and resume using the cluster. If you shut down your cluster but you don't create a final snapshot, the cluster will be deleted. In either case, automated snapshots are deleted after the cluster is shut down, but any manual snapshots are retained until you delete them. You might be charged Amazon Simple Storage Service storage rates for manual snapshots, depending on the amount of storage you have available for Amazon Redshift snapshots for your clusters. For more information, see Shutting Down and Deleting Clusters (p. 12).

To shut down or delete a cluster

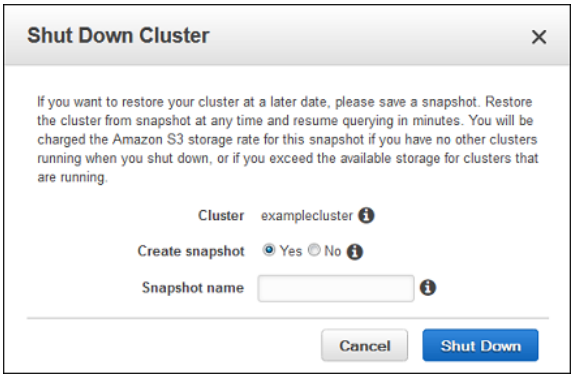
1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster that you want to shut down or delete.

3. On the **Configuration** tab of the cluster details page, click **Cluster**, and then click **Shut Down**.

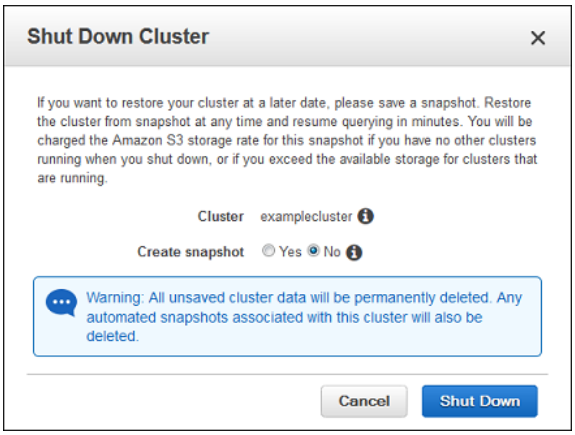


4. In the **Shut Down Clusters** dialog box, do one of the following:

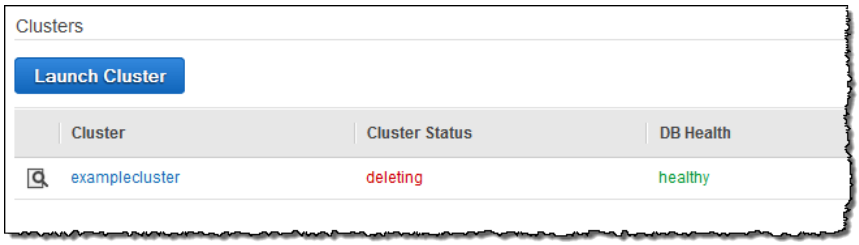
   - In **Create snapshot**, click **Yes** to shut down the cluster and take a final snapshot. In **Snapshot name**, type a name for the final snapshot, and then click **Shut Down**.


Shut Down Cluster
If you want to restore your cluster at a later date, please save a snapshot. Restore the cluster from snapshot at any time and resume querying in minutes. You will be charged the Amazon S3 storage rate for this snapshot if you have no other clusters running when you shut down, or if you exceed the available storage for clusters that are running.
Cluster
examplecluster
Create snapshot
Yes
No
Snapshot name
Cancel
Shut Down

   - In **Create snapshot**, click **No** to shut down and delete the cluster without taking a final snapshot, and then click **Shut Down**.


Shut Down Cluster
If you want to restore your cluster at a later date, please save a snapshot. Restore the cluster from snapshot at any time and resume querying in minutes. You will be charged the Amazon S3 storage rate for this snapshot if you have no other clusters running when you shut down, or if you exceed the available storage for clusters that are running.
Cluster examplecluster
Create snapshot Yes No
Warning: All unsaved cluster data will be permanently deleted. Any automated snapshots associated with this cluster will also be deleted.
Cancel
Shut Down

After you initiate the shutdown of the cluster, it can take several minutes for the cluster to be shut down. You can monitor the status in the cluster list as shown in the following screenshot. If you requested a final snapshot, **Cluster Status** will show `final-snapshot` before `deleting`.



When the process has finished, you can verify that the cluster has been shut down because it will no longer appear in the list of clusters on the **Clusters** page.

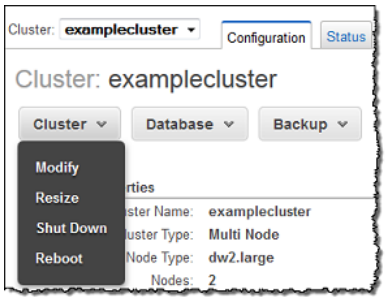
Rebooting a Cluster

When you reboot a cluster, the cluster status is set to `rebooting` and a cluster event is created when the reboot is completed. Any pending cluster modifications are applied at this reboot.

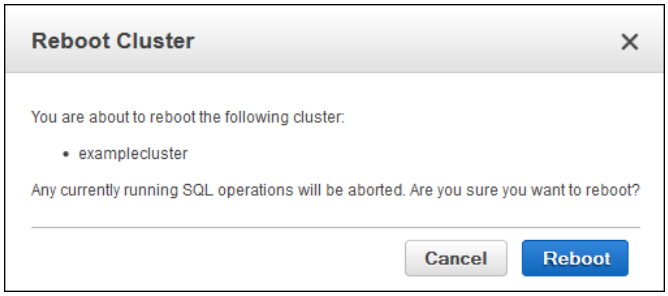
To reboot a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster that you want to reboot.

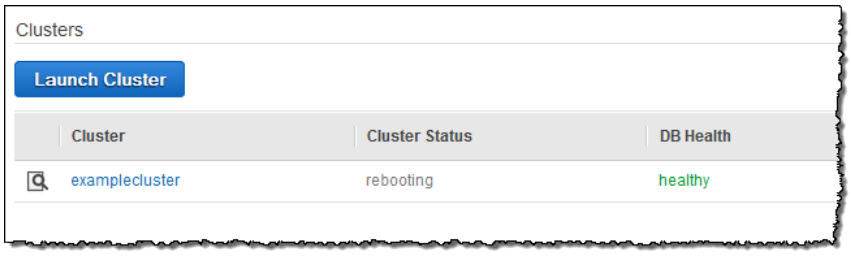
3. On the **Configuration** tab of the cluster details page, click **Cluster** and then click **Reboot**.



4. In the **Reboot Clusters** window, confirm that you want to reboot this cluster, and then click **Reboot**.


Reboot Cluster
You are about to reboot the following cluster:
examplecluster
Any currently running SQL operations will be aborted. Are you sure you want to reboot?
Cancel
Reboot

It can take several minutes for the cluster to be available. You can monitor the status of the reboot in the cluster list as shown in the following screenshot.



Resizing a Cluster

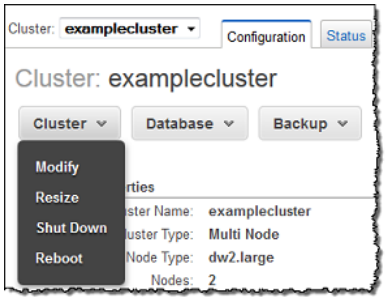
When you resize a cluster, you specify a number of nodes or node type that is different from the current configuration of the cluster. While the cluster is in the process of resizing, you cannot run any write or read/write queries on the cluster; you can run only read queries.

For more information about resizing clusters, including walking through the process of resizing clusters using different approaches, see Tutorial: Resizing Clusters in Amazon Redshift (p. 224).

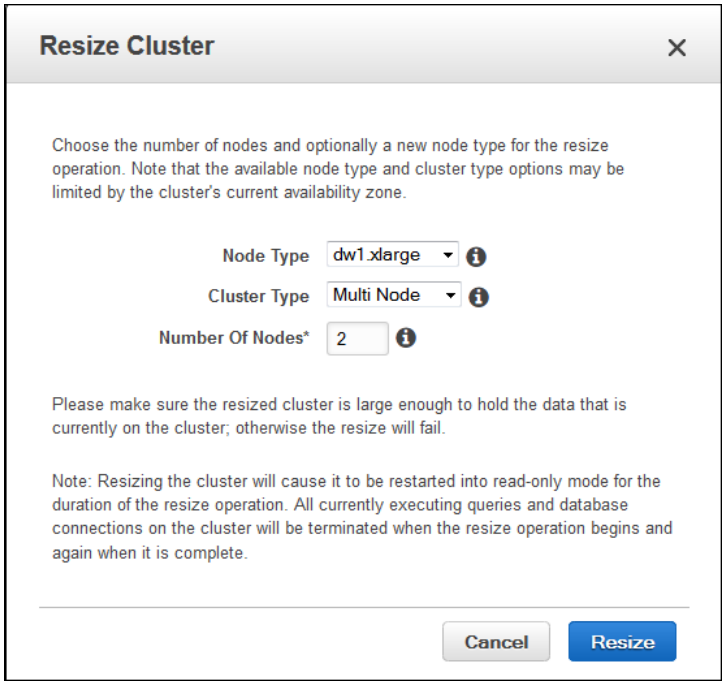
To resize a cluster

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster that you want to resize.

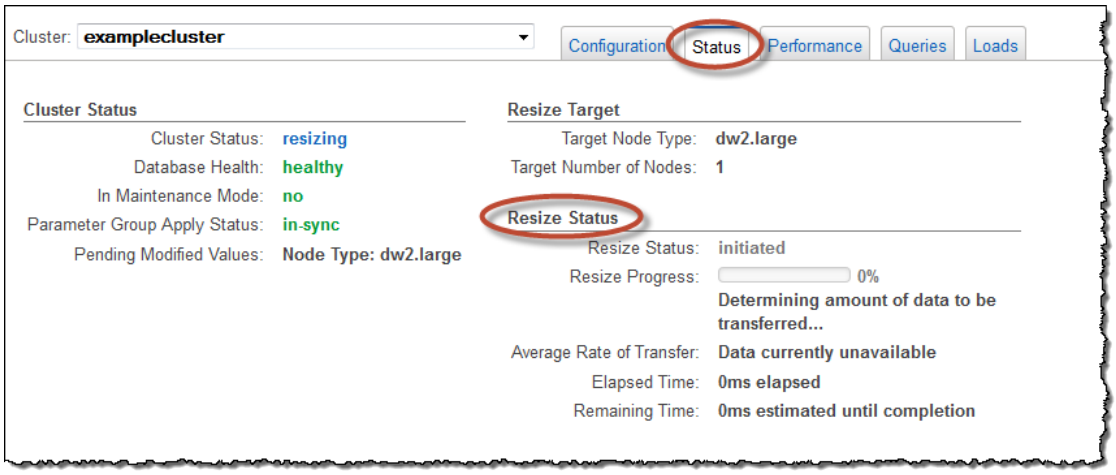
3. On the **Configuration** tab of the cluster details page, click **Cluster** and then click **Resize**.



4. In the **Resize Clusters** window, configure the resize parameters including the **Node Type**, **Cluster Type**, and **Number of Nodes**, and then click **Resize**.


Resize Cluster
Choose the number of nodes and optionally a new node type for the resize operation. Note that the available node type and cluster type options may be limited by the cluster's current availability zone.
Node Type
dw1.xlarge
Cluster Type
Multi Node
Number Of Nodes*
2
Please make sure the resized cluster is large enough to hold the data that is currently on the cluster; otherwise the resize will fail.
Note: Resizing the cluster will cause it to be restarted into read-only mode for the duration of the resize operation. All currently executing queries and database connections on the cluster will be terminated when the resize operation begins and again when it is complete.
Cancel
Resize

You can monitor the progress of the resize on the **Status** tab.



Getting Information About Cluster Configuration

To get cluster configuration details

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster for which you want to view configuration information.

3. On the **Configuration** tab of the cluster details page, review the configuration information. You can view information about the cluster properties, status, database, capacity, backup, audit logging, maintenance, and SSH ingestion settings.

Getting an Overview of Cluster Status

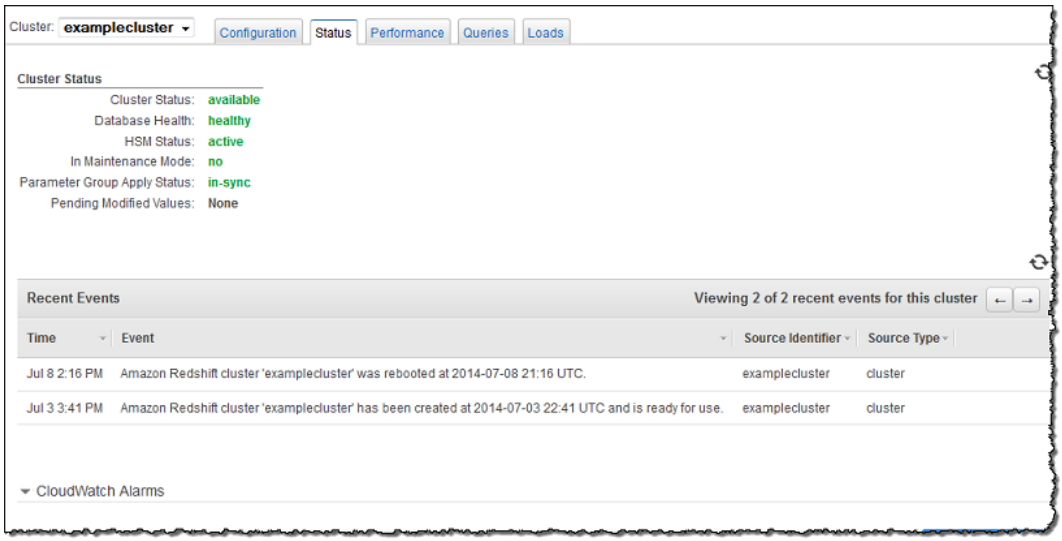
The cluster **Status** tab provides a high level overview of the status of a cluster, a summary of events related to the cluster, and a list of Amazon CloudWatch alarms associated with the cluster.

To get an overview of cluster status

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster for which you want to view status information.

3. Click the **Status** tab.

   The status summary page is displayed as shown in the following screenshot.



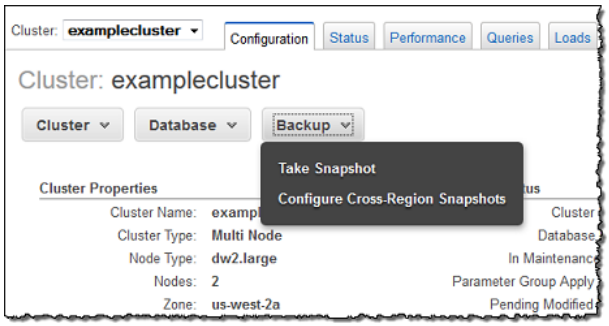
Taking a Snapshot of a Cluster

You can take a snapshot of your cluster from the **Configuration** tab of your cluster as shown following. You can also take a snapshot of your cluster from the snapshots part of the Amazon Redshift console. For more information, go to Managing Snapshots Using the Console (p. 86).

To take a snapshot of a cluster

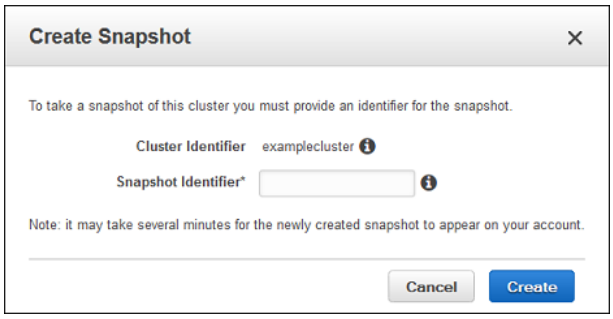
1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster for which you want to take a snapshot.

3. On the **Configuration** tab of the cluster details page, click **Backup**, and then click **Take Snapshot**.



4. In the **Create Snapshot** dialog box, do the following:

   a. In the **Cluster Identifier** box, click the cluster that you want to take a snapshot of.
   b. In the **Snapshot Identifier** box, type a name for the snapshot.


Create Snapshot
To take a snapshot of this cluster you must provide an identifier for the snapshot.
Cluster Identifier
examplecluster
Snapshot Identifier*
Note: it may take several minutes for the newly created snapshot to appear on your account.
Cancel
Create

5. Click **Create**.

   To view details about the snapshot taken and all other snapshots for your AWS account, go to the snapshots part of the Amazon Redshift console (see Managing Snapshots Using the Console (p. 86)).

Editing the Default Disk Space Alarm

If you opted to create a default disk space alarm when you created your Amazon Redshift cluster, you can edit the alarm. For example, you might want to change the percentage at which the alarm triggers, or you might want to change the duration settings.

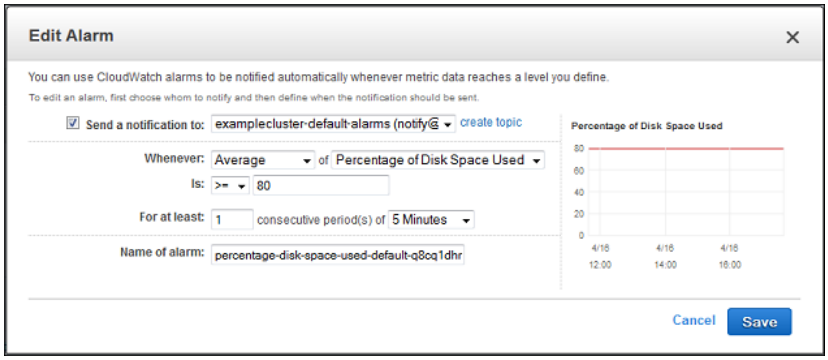
To edit the default disk space alarm

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**, and then click the cluster associated with the alarm that you want to edit.
3. Click the **Status** tab.
4. In the **CloudWatch Alarms** section, click the alarm that you want to edit.

   The default disk space alarm that was created when you launched your cluster is named **percentage-disk-space-used-default-<*string*>.** The *string* is randomly generated by Amazon Redshift.

CloudWatch Alarms
1 of 1 in OK
Create Alarm
Below are your CloudWatch alarms for the selected resources. Click on an alarm to edit it or click 'view' to see additional options and details in Amazon CloudWatch.
View all CloudWatch alarms
State
Name
More Options
OK
percentage-disk-space-used-default-nclfunppbhmu
view

5. In the **Edit Alarm** window, edit any values that you want to change, such as the percentage or minutes.



6. To change the Amazon SNS topic that the alarm is associated with, do one of the following:

   - If you want to select another existing topic, select a topic from the **Send a notification to** list.
   - If you want to create a new topic, click **create topic** and specify a new topic name and the email addresses for recipients.

7. Click **Save**.

Working with Cluster Performance Data

You can work with cluster performance data using the **Performance**, **Queries**, and **Loads** tabs. For more information about working with cluster performance, see Working with Performance Data in the Amazon Redshift Console (p. 183).

Managing Clusters Using the AWS SDK for Java

The following Java code example demonstrates common cluster management operations including:

- Creating a cluster.
- Listing metadata about a cluster.
- Modifying configuration options.

After you initiate the request for the cluster to be created, you must wait until the cluster is in the `available` state before you can modify it. This example uses a loop to periodically check the status of the cluster using the `describeClusters` method. When the cluster is available, the preferred maintenance window for the cluster is changed.

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and specify a cluster identifier.

```
import java.io.IOException;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyCluster {

    public static AmazonRedshiftClient client;
    public static String clusterIdentifier = "***provide a cluster identifi
er***";
    public static long sleepTime = 20;

    public static void main(String[] args) throws IOException {

        AWSCredentials credentials = new PropertiesCredentials(
                CreateAndModifyCluster.class
                        .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             createCluster();
             waitForClusterReady();
             describeClusters();
             modifyCluster();
             describeClusters();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createCluster() {
       CreateClusterRequest request = new CreateClusterRequest()
          .withClusterIdentifier(clusterIdentifier)
          .withMasterUsername("masteruser")
          .withMasterUserPassword("12345678Aa")
          .withNodeType("dw.hs1.xlarge")
          .withNumberOfNodes(2);

       Cluster createResponse = client.createCluster(request);
      System.out.println("Created cluster " + createResponse.getClusterIdenti
fier());
    }

    private static void describeClusters() {
       DescribeClustersRequest request = new DescribeClustersRequest()
          .withClusterIdentifier(clusterIdentifier);

       DescribeClustersResult result = client.describeClusters(request);
       printResult(result);
    }

    private static void modifyCluster() {
```

```
        ModifyClusterRequest request = new ModifyClusterRequest()
           .withClusterIdentifier(clusterIdentifier)
           .withPreferredMaintenanceWindow("wed:07:30-wed:08:00");

        client.modifyCluster(request);
        System.out.println("Modified cluster " + clusterIdentifier);

    }

    private static void printResult(DescribeClustersResult result)
    {
        if (result == null)
        {
            System.out.println("Describe clusters result is null.");
            return;
        }

        System.out.println("Cluster property:");
        System.out.format("Preferred Maintenance Window: %s\n", res
ult.getClusters().get(0).getPreferredMaintenanceWindow());
    }

    private static void waitForClusterReady() throws InterruptedException {
        Boolean clusterReady = false;
        System.out.println("Wating for cluster to become available.");
        while (!clusterReady) {
          DescribeClustersResult result = client.describeClusters(new Describe
ClustersRequest()
                .withClusterIdentifier(clusterIdentifier));
            String status = (result.getClusters()).get(0).getClusterStatus();
            if (status.equalsIgnoreCase("available")) {
                clusterReady = true;
            }
            else {
                System.out.print(".");
                Thread.sleep(sleepTime*1000);
            }
        }
    }
}
```

Manage Clusters Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage clusters.

- create-cluster
- delete-cluster
- describe-clusters
- describe-cluster-versions
- describe-orderable-cluster-options
- modify-cluster
- reboot-cluster

You can use the following Amazon Redshift APIs to manage clusters.

- CreateCluster
- DeleteCluster
- DescribeClusters
- DescribeClusterVersions
- DescribeOrderableClusterOptions
- ModifyCluster
- RebootCluster

Managing Clusters in an Amazon Virtual Private Cloud (VPC)

Topics

- Overview (p. 32)
- Creating a Cluster in a VPC (p. 33)
- Managing VPC Security Groups for a Cluster (p. 35)
- Amazon Redshift Cluster Subnet Groups (p. 36)

Overview

Amazon Redshift supports both the EC2-VPC and EC2-Classic platforms to launch a cluster. For more information, see Supported Platforms to Launch Your Cluster (p. 9).

When provisioning a cluster in VPC, you need to do the following:

- **Provide VPC information.**

  When you request Amazon Redshift to create a cluster in your VPC, you must provide your VPC information, such as the VPC ID, and a list of subnets in your VPC by first creating a cluster subnet group. When you launch a cluster you provide the cluster subnet group so that Amazon Redshift can provision your cluster in one of the subnets in the VPC. For more information about creating subnet groups in Amazon Redshift, see Amazon Redshift Cluster Subnet Groups (p. 36). For more information about setting up VPC, go to Getting Started with Amazon VPC in the *VPC Amazon Virtual Private Cloud Getting Started Guide*.
- **Optionally, configure the publicly accessible options.**

  If you configure your cluster to be publicly accessible, you can optionally select an *elastic IP address (EIP)* to use for the external IP address. An EIP is a static IP address that is associated with your AWS account. You can use an EIP to connect to your cluster from outside the VPC. An EIP gives you the ability to change your underlying configuration without affecting the IP address that clients use to connect to your cluster. This approach can be helpful for situations such as recovery after a failure.

  If you want to use an EIP associated with your own AWS account, you must create it in Amazon EC2 prior to launching your Amazon Redshift cluster. Otherwise, it will not be available during the launch process. You can also have Amazon Redshift configure an EIP to use for the VPC, but the assigned EIP will be managed by the Amazon Redshift service and will not be associated with your AWS account. For more information, go to Elastic IP Addresses (EIP) in the *Amazon EC2 User Guide for Linux Instances*.

If you have a publicly accessible cluster in a VPC, and you want to connect to it by using the private IP address from within the VPC, you must set the following VPC parameters to *true*:

- *DNS resolution*
- *DNS hostnames*

If you have a publicly accessible cluster in a VPC, but do not set those parameters to *true* in the VPC, connections made from within the VPC will resolve to the EIP of the cluster instead of the private IP address. We recommend that you set these parameters to *true* and use the private IP address for a publicly accessible cluster when connecting from within the VPC. For more information, see Using DNS with Your VPC in the Amazon VPC User Guide.

Note
If you have an existing publicly accessible cluster in a VPC, connections from within the VPC will continue to use the EIP to connect to the cluster even with those parameters set until you resize the cluster. Any new clusters will follow the new behavior of using the private IP address when connecting to the publicly accessible cluster from within the same VPC.

Also, note that the *EIP* is an external IP address for accessing the cluster outside of a VPC, but it is not related to the *cluster node public IP addresses and private IP addresses* that are displayed in the Amazon Redshift console under **SSH Ingestion Settings**. The public and private cluster node IP addresses appear regardless of whether the cluster is publicly accessible or not. They are used only in certain circumstances to configure ingress rules on the remote host when you load data from an Amazon EC2 instance or other remote host using a Secure Shell (SSH) connection. For more information, see Step 1: Retrieve the cluster public key and cluster node IP addresses in the Amazon Redshift Database Developer Guide.

The option to associate a cluster with an EIP is available only when you create the cluster or restore the cluster from a snapshot. You can't attach an EIP after the cluster is created or restored. If you want to associate the cluster with an EIP or change an EIP that is associated with the cluster, you need to restore the cluster from a snapshot and specify the EIP at that time.

- **Associate a VPC security group.**

  You then grant inbound access using a VPC security group. This VPC security group must allow access over the database port for the cluster so that you can connect by using SQL client tools. You can configure this in advance, or add rules to it after you launch the cluster. For more information, go to Security in Your VPC in the *Amazon Virtual Private Cloud User Guide*. You cannot use the Amazon Redshift cluster security groups to grant inbound access to the cluster.

For more information about working with clusters in a VPC, see Creating a Cluster in a VPC (p. 33).

Restoring a Snapshot of a Cluster in VPC

A snapshot of a cluster in VPC can only be restored in a VPC, not outside the VPC. You can restore it in the same VPC or another VPC in your account. For more information about snapshots, see Amazon Redshift Snapshots (p. 82).

Creating a Cluster in a VPC

The following are the general steps how you can deploy a cluster in your VPC.

To create a cluster in a VPC

1. Set up a VPC.

   You can create your cluster either in the default VPC for your account, if your account has one, or a VPC that you have created. For more information, see Supported Platforms to Launch Your Cluster (p. 9). To create a VPC, follow steps 2 and 3 in the Amazon Virtual Private Cloud Getting

Started Guide. Make a note of the VPC identifier, subnet, and subnet's availability zone. You will need this information when you launch your cluster.

Note
You must have at least one subnet defined in your VPC so you can add it to the cluster subnet group in the next step. If you use the VPC Wizard, a subnet for your VPC is automatically created for you. For more information about adding a subnet to your VPC, go to Adding a Subnet to Your VPC.

2. Create an Amazon Redshift cluster subnet group that specifies which of the subnets in the VPC can be used by the Amazon Redshift cluster.

   You can create cluster subnet group using either the Amazon Redshift console or programmatically. For more information, see Amazon Redshift Cluster Subnet Groups (p. 36).

3. Authorize access for inbound connections in a VPC security group that you will associate with the cluster.

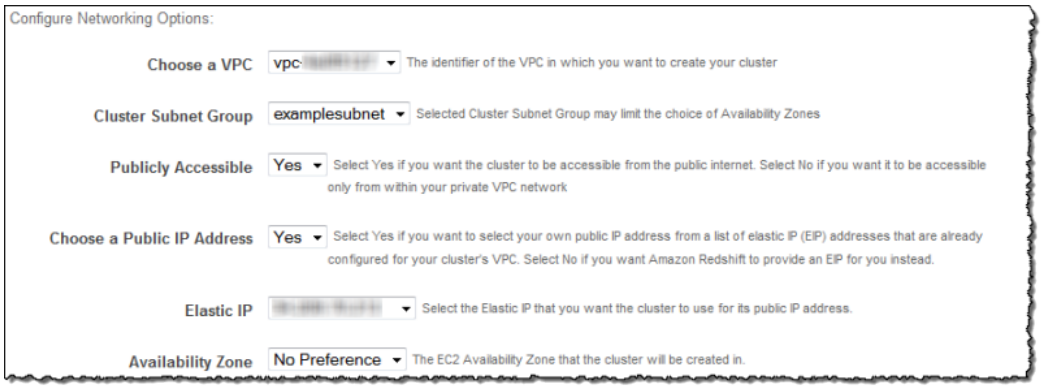
   To enable a client outside the VPC (on the public Internet) to connect to the cluster, you must associate the cluster with a VPC security group that grants inbound access to the port that you used when you launched the cluster. For examples of security group rules, go to Security Group Rules in the *Amazon Virtual Private Cloud User Guide*.

4. Launch a cluster in your VPC.

   You can use the procedure described in the Getting Started to launch the cluster in your VPC. For more information, see Step 2: Launch a Cluster. As you follow the wizard, in the **Configure Network Options** of the **ADDITIONAL CONFIGURATION** page, specify the following information:

   - **Choose a VPC** Select the VPC from the drop-down list.
   - **Cluster Subnet Group** Select the cluster subnet group you created in step 2.
   - **Publicly Accessible** Select **Yes** if you want the cluster to have a public IP address that can be accessed from the public internet, select **No** if you want the cluster to have a private IP addressed that can only be accessed from within the VPC. If your AWS account allows you to create EC2-Classic clusters, the default is no, otherwise the default is yes.
   - **Choose a Public IP Address** Select **Yes** if you want to select an elastic IP (EIP) address that you already have configured. Otherwise, select **No** to have Amazon Redshift create an EIP for your instance.
   - **Elastic IP** Select an EIP to use to connect to the cluster from outside of the VPC.
   - **Availability Zone** Select **No Preference** to have Amazon Redshift select the availability zone that the cluster will be created in. Otherwise, select a specific availability zone.
   - Select the VPC security group that grants authorized devices access to the cluster.

   The following is an example screen shot of the **Configure Networking Options** section of the **ADDITIONAL CONFIGURATION** page.


Configure Networking Options:
Choose a VPC
vpc
The identifier of the VPC in which you want to create your cluster
Cluster Subnet Group
examplesubnet
Selected Cluster Subnet Group may limit the choice of Availability Zones
Publicly Accessible
Yes
Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network
Choose a Public IP Address
Yes
Select Yes if you want to select your own public IP address from a list of elastic IP (EIP) addresses that are already configured for your cluster's VPC. Select No if you want Amazon Redshift to provide an EIP for you instead.
Elastic IP
Select the Elastic IP that you want the cluster to use for its public IP address.
Availability Zone
No Preference
The EC2 Availability Zone that the cluster will be created in.

Now you are ready to use the cluster. You can follow the Getting Started steps to test the cluster by uploading sample data and trying example queries.

Managing VPC Security Groups for a Cluster

When you provision an Amazon Redshift cluster, it is locked down by default so nobody has access to it. To grant other users inbound access to an Amazon Redshift cluster, you associate the cluster with a security group. If you are on the EC2-VPC platform, you can either use an existing Amazon VPC security group or define a new one and then associate it with a cluster as described following. If you are on the EC2-Classic platform, you define a cluster security group and associate it with a cluster. For more information on using cluster security groups on the EC2-Classic platform, see Amazon Redshift Cluster Security Groups (p. 43).

A VPC security group consists of a set of rules that control access to an instance on the VPC, such as your cluster. Individual rules set access based either on ranges of IP addresses or on other VPC security groups. When you associate a VPC security group with a cluster, the rules that are defined in the VPC security group control access to the cluster.

Each cluster you provision on the EC2-VPC platform has one or more Amazon VPC security groups associated with it. Amazon VPC provides a VPC security group called default, which is created automatically when you create the VPC. Each cluster that you launch in the VPC is automatically associated with the default VPC security group if you don't specify a different VPC security group when you create the cluster. You can associate a VPC security group with a cluster when you create the cluster, or you can associate a VPC security group later by modifying the cluster. For more information on associating a VPC security group with a cluster, see To create a cluster (p. 14) and To modify a cluster (p. 21).

The following table describes the default rules for the default VPC security group.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The security group ID (sg-*xxxxxxxx*)	All	All	Allow inbound traffic from instances assigned to the same security group
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	All	All	Allow all outbound traffic

You can change the rules for the default VPC security group as needed for your Amazon Redshift cluster.

If the default VPC security group is enough for you, you don’t need to create more. However, you can optionally create additional VPC security groups to better manage inbound access to your cluster. For example, suppose you are running a service on an Amazon Redshift cluster, and you have several different service levels you provide to your customers. If you don’t want to provide the same access at

all service levels, you might want to create separate VPC security groups, one for each service level. You can then associate these VPC security groups with your cluster.

Keep in mind that while you can create up to 100 VPC security groups for a VPC, and you can associate a VPC security group with many clusters, you can only associate up to 5 VPC security groups with a given cluster.

Amazon Redshift applies changes to a VPC security group immediately. So if you have associated the VPC security group with a cluster, inbound cluster access rules in the updated VPC security group apply immediately.

You can create and modify VPC security groups in the AWS VPC console. You can also manage VPC security groups programmatically by using the AWS CLI, the AWS EC2 CLI, and the AWS Tools for Windows PowerShell. For more information about working with VPC security groups, go to Security Groups for Your VPC in the *Amazon Virtual Private Cloud User Guide*.

Amazon Redshift Cluster Subnet Groups

Topics

- Overview (p. 36)
- Managing Cluster Subnet Groups Using the Console (p. 36)
- Managing Cluster Subnet Groups Using the AWS SDK for Java (p. 39)
- Manage Cluster Subnet Groups Using Amazon Redshift CLI and API (p. 42)

Overview

You create a cluster subnet group if you are provisioning your cluster in your virtual private cloud (VPC). For more information about VPC, go to Amazon Virtual Private Cloud (Amazon VPC) *product detail page*.

Your VPC can have one or more subnets, a subset of IP addresses within your VPC, that enable you to group your resources based on your security and operation needs. A cluster subnet group allows you to specify a set of subnets in your VPC. When provisioning a cluster you provide the subnet group and Amazon Redshift creates the cluster on one of the subnets in the group.

For more information about creating a VPC, go to Amazon Virtual Private Cloud (VPC) Documentation.

After creating a subnet group, you can remove subnets you previously added or add more subnets. Amazon Redshift provides APIs for you to create, modify or delete a cluster subnet group. You can also perform these operations in the console.

Managing Cluster Subnet Groups Using the Console

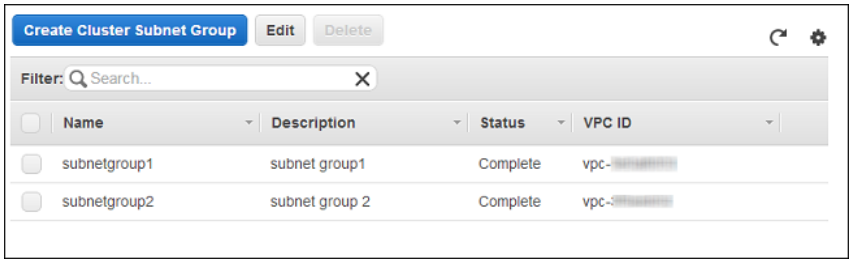
Topics

- Creating a Cluster Subnet Group (p. 37)
- Modifying a Cluster Subnet Group (p. 38)
- Deleting a Cluster Subnet Group (p. 39)

The section explains how to manage your cluster subnet groups using the Amazon Redshift console. You can create a cluster subnet group, manage an existing one, or delete one. All of these tasks start from the cluster subnet group list. You must select a cluster subnet group to manage it.

In the example cluster subnet group list below, there is one cluster subnet group. By default, there are no cluster subnet groups defined for your AWS account. Because `my-subnet-group` is selected

(highlighted), you can edit or delete it. The details of the selected security group are shown under **Cluster Subnet Group Details**.

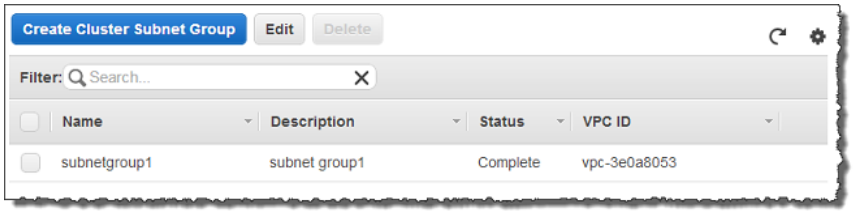


Creating a Cluster Subnet Group

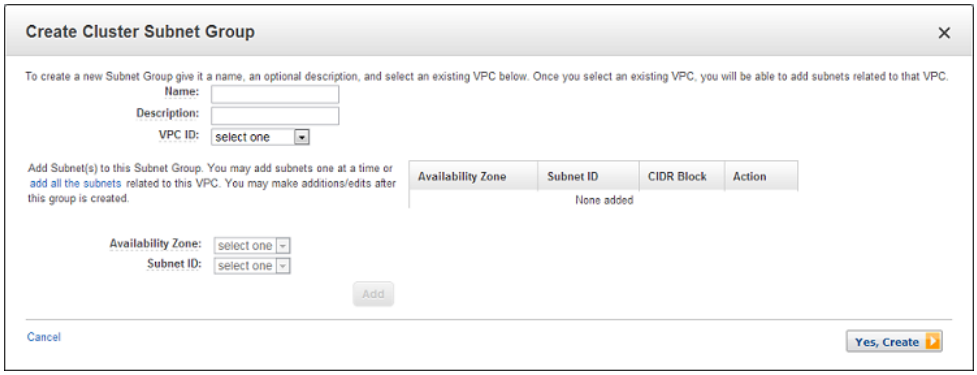
You must have at least one cluster subnet group defined to provision a cluster in a VPC.

To create a cluster subnet group

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Subnet Groups** tab, click **Create Cluster Subnet Group**.



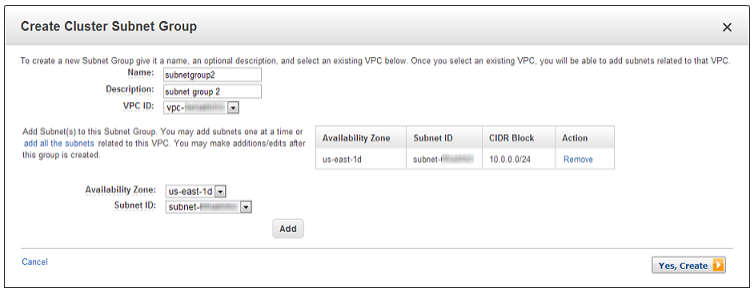
4. In the **Create Cluster Subnet Group** dialog box, add subnets to the group.


Create Cluster Subnet Group
To create a new Subnet Group give it a name, an optional description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.
Name:
Description:
VPC ID:
select one
Add Subnet(s) to this Subnet Group. You may add subnets one at a time or add all the subnets related to this VPC. You may make additions/edits after this group is created.
Availability Zone
Subnet ID
CIDR Block
Action
None added
Availability Zone:
Subnet ID:
select one
Add
Cancel
Yes, Create

   a. Specify a **Name**, **Description**, and **VPC ID** for the cluster subnet group.
   b. Add subnets to the group by doing one of the following:

      - Click **add all the subnets** link. or
      - Use the **Availability Zone** and **Subnet ID** boxes to choose a specific subnet and then click **Add**.

      The following example shows a cluster subnet group specified with one subnet group.



5. Click **Yes, Create**.

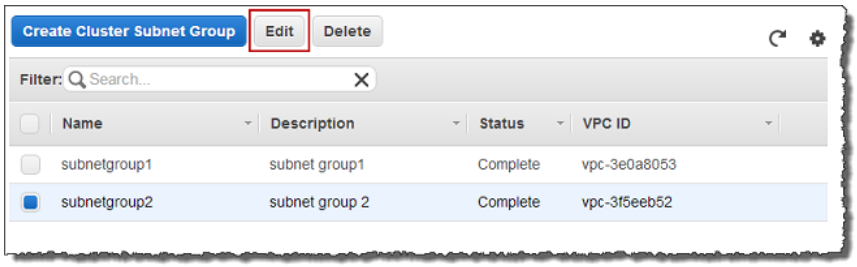
   The new group will be displayed in the list of cluster subnet groups.

Modifying a Cluster Subnet Group

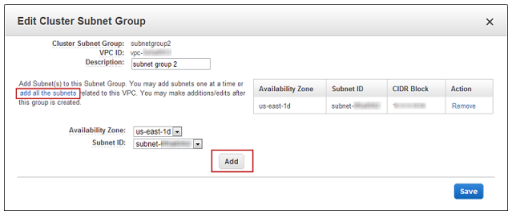
To modify a cluster subnet group

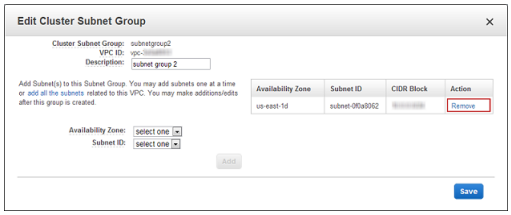
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Subnet Groups** tab, in the cluster subnet group list, click the row of the group you want to modify, and then click **Edit**.

   In the example below, `subnetgroup2` is the cluster subnet group we want to modify.



4. In the **Cluster Subnet Group Details**, take one of the following actions.

To...	Do this...
Add one or more subnets to the group.	Select an individual subnet by using the **Availability Zone** and **Subnet ID** boxes or click **add all the subnets**.  Click **Save**.

To...	Do this...
Remove a subnet from the group.	In the lists of subnets in use for the group, click **Remove** next to the subnet to remove.  Click **Save**.

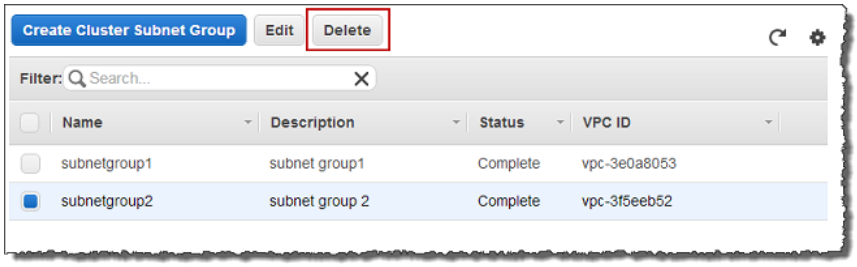
Deleting a Cluster Subnet Group

You cannot delete a cluster subnet group that is used by a cluster.

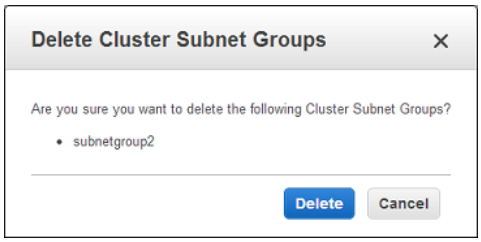
To delete a cluster subnet group

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Subnet Groups** tab, in the cluster subnet group list, click the row of the group you want to delete.

   In the example below, `my-subnet-group` is the cluster subnet group we want to delete.

4. In the Delete Cluster Subnet Group dialog box, click **Delete**.

Managing Cluster Subnet Groups Using the AWS SDK for Java

The following Java code example demonstrates common cluster subnet operations including:

- Creating a cluster subnet group.
- Listing metadata about a cluster subnet group.
- Modifying a cluster subnet group.

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and provide a cluster subnet group name and two subnet identifiers.

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterSubnetGroup {

    public static AmazonRedshiftClient client;
    public static String clusterSubnetGroupName = "***provide a cluster subnet
 group name ****";
    // You can use the VPC console to find subnet IDs to use.
    public static String subnetId1 = "***provide a subnet ID****";
    public static String subnetId2 = "***provide a subnet ID****";

    public static void main(String[] args) throws IOException {

        AWSCredentials credentials = new PropertiesCredentials(
                CreateAndModifyClusterSubnetGroup.class
                        .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             createClusterSubnetGroup();
             describeClusterSubnetGroups();
             modifyClusterSubnetGroup();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createClusterSubnetGroup() {
        CreateClusterSubnetGroupRequest request = new CreateClusterSubnet
GroupRequest()
            .withClusterSubnetGroupName(clusterSubnetGroupName)
            .withDescription("my cluster subnet group")
            .withSubnetIds(subnetId1);
        client.createClusterSubnetGroup(request);
       System.out.println("Created cluster subnet group: " + clusterSubnetGroup
Name);
    }

    private static void modifyClusterSubnetGroup() {
        // Get existing subnet list.
```

```
       DescribeClusterSubnetGroupsRequest request1 = new DescribeClusterSubnet
GroupsRequest()
            .withClusterSubnetGroupName(clusterSubnetGroupName);
        DescribeClusterSubnetGroupsResult result1 = client.describeClusterSub
netGroups(request1);
        List<String> subnetNames = new ArrayList<String>();
        // We can work with just the first group returned since we requested
info about one group.
        for (Subnet subnet : result1.getClusterSubnetGroups().get(0).getSub
nets()) {
            subnetNames.add(subnet.getSubnetIdentifier());
        }
        // Add to existing subnet list.
        subnetNames.add(subnetId2);

        ModifyClusterSubnetGroupRequest request = new ModifyClusterSubnet
GroupRequest()
            .withClusterSubnetGroupName(clusterSubnetGroupName)
            .withSubnetIds(subnetNames);
        ClusterSubnetGroup result2 = client.modifyClusterSubnetGroup(request);

        System.out.println("\nSubnet group modified.");
        printResultSubnetGroup(result2);
    }


    private static void describeClusterSubnetGroups() {
       DescribeClusterSubnetGroupsRequest request = new DescribeClusterSubnet
GroupsRequest()
        .withClusterSubnetGroupName(clusterSubnetGroupName);

    DescribeClusterSubnetGroupsResult result = client.describeClusterSubnet
Groups(request);
    printResultSubnetGroups(result);
    }

   private static void printResultSubnetGroups(DescribeClusterSubnetGroupsResult
 result)
    {
        if (result == null)
        {
            System.out.println("\nDescribe cluster subnet groups result is
null.");
            return;
        }

        for (ClusterSubnetGroup group : result.getClusterSubnetGroups())
        {
            printResultSubnetGroup(group);
        }

    }
    private static void printResultSubnetGroup(ClusterSubnetGroup group) {
       System.out.format("Name: %s, Description: %s\n", group.getClusterSubnet
GroupName(), group.getDescription());
        for (Subnet subnet : group.getSubnets()) {
           System.out.format("  Subnet: %s, %s, %s\n", subnet.getSubnetIdenti
fier(),
```

```
                    subnet.getSubnetAvailabilityZone().getName(), subnet.getSub
netStatus());
            }
        }
}
```

Manage Cluster Subnet Groups Using Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage cluster subnet groups.

- create-cluster-subnet-group
- delete-cluster-subnet-group
- describe-cluster-subnet-groups
- modify-cluster-subnet-group

You can use the following Amazon Redshift APIs to manage cluster subnet groups.

- CreateClusterSubnetGroup
- DeleteClusterSubnetGroup
- DescribeClusterSubnetGroups
- ModifyClusterSubnetGroup

Amazon Redshift Cluster Security Groups

When you provision an Amazon Redshift cluster, it is locked down by default so nobody has access to it. To grant other users inbound access to an Amazon Redshift cluster, you associate the cluster with a security group. If you are on the EC2-Classic platform, you define a cluster security group and associate it with a cluster as described following. If you are on the EC2-VPC platform, you can either use an existing Amazon VPC security group or define a new one and then associate it with a cluster. For more information on managing a cluster on the EC2-VPC platform, see Managing Clusters in an Amazon Virtual Private Cloud (VPC) (p. 32).

Topics

- Overview (p. 43)
- Managing Cluster Security Groups Using the Console (p. 44)
- Managing Cluster Security Groups Using the AWS SDK for Java (p. 52)
- Manage Cluster Security Groups Using the Amazon Redshift CLI and API (p. 55)

Overview

A cluster security group consists of a set of rules that control access to your cluster. Individual rules identify either a range of IP addresses or an Amazon EC2 security group that is allowed access to your cluster. When you associate a cluster security group with a cluster, the rules that are defined in the cluster security group control access to the cluster.

You can create cluster security groups independent of provisioning any cluster. You can associate a cluster security group with an Amazon Redshift cluster either at the time you provision the cluster or later. Also, you can associate a cluster security group with multiple clusters.

Amazon Redshift provides a cluster security group called default, which is created automatically when you launch your first cluster. Initially, this cluster security group is empty. You can add inbound access rules to the default cluster security group and then associate it with your Amazon Redshift cluster.

If the default cluster security group is enough for you, you don't need to create your own. However, you can optionally create your own cluster security groups to better manage inbound access to your cluster. For example, suppose you are running a service on an Amazon Redshift cluster, and you have a few companies as your customers. If you don't want to provide the same access to all your customers, you

might want to create separate cluster security groups, one for each company. You can add rules in each cluster security group to identify the Amazon EC2 security groups and the IP address ranges specific to a company. You can then associate all these cluster security groups with your cluster.

You can associate a cluster security group with many clusters, and you can associate many cluster security groups with a cluster.

You can manage cluster security groups using the Amazon Redshift console, and you can manage cluster security groups programmatically by using the Amazon Redshift API or the AWS SDKs.

Amazon Redshift applies changes to a cluster security group immediately. So if you have associated the cluster security group with a cluster, inbound cluster access rules in the updated cluster security group apply immediately.

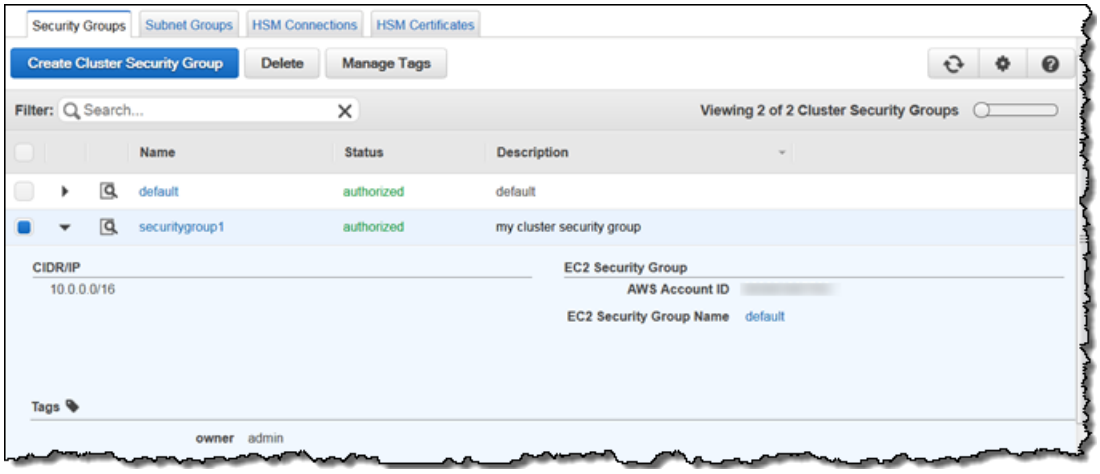
Managing Cluster Security Groups Using the Console

Topics

- Creating a Cluster Security Group (p. 45)
- Tagging a Cluster Security Group (p. 46)
- Managing Ingress Rules for a Cluster Security Group (p. 47)
- Revoking Ingress Rules for a Cluster Security Group (p. 49)
- Tagging Ingress Rules for a Cluster Security Group (p. 49)
- Deleting a Cluster Security Group (p. 51)
- Associating a Cluster Security Group with a Cluster (p. 52)

You can create, modify, and delete cluster security groups by using the Amazon Redshift console. You can also manage the default cluster security group in the Amazon Redshift console. All of the tasks start from the cluster security group list. You must select a cluster security group to manage it.

In the example cluster security group list below, there are two cluster security groups, the `default` cluster security group and a custom cluster security group called `securitygroup1`. Because `securitygroup1` is selected (highlighted), you can delete it or manage tags for it, and also see the rules and tags associated with it.



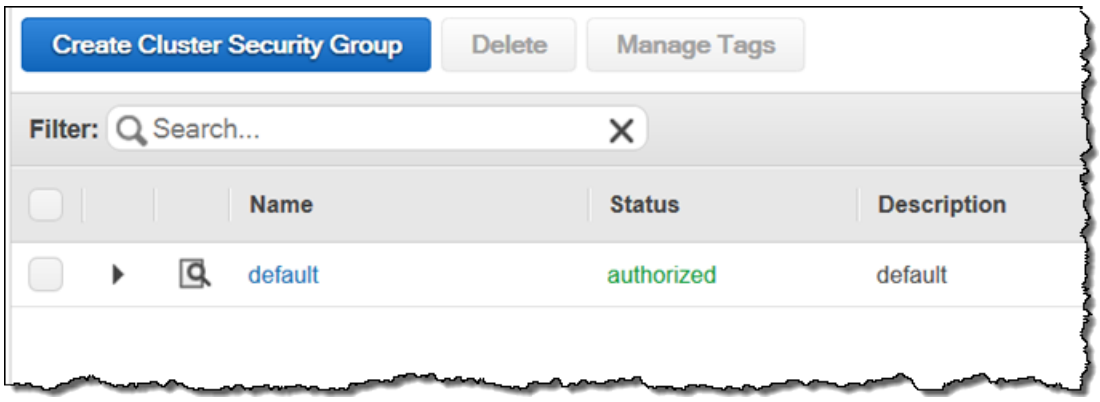
You cannot delete the default cluster security group, but you can modify it by authorizing or revoking ingress access.

To add or modify the rules associated with a security group, click on the security group to go to the **Security Group Connections** page.

Creating a Cluster Security Group

To create a cluster security group

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Security Groups** tab, click **Create Cluster Security Group**.



4. In the **Create Cluster Security Group** dialog box, specify a cluster security group name and description.

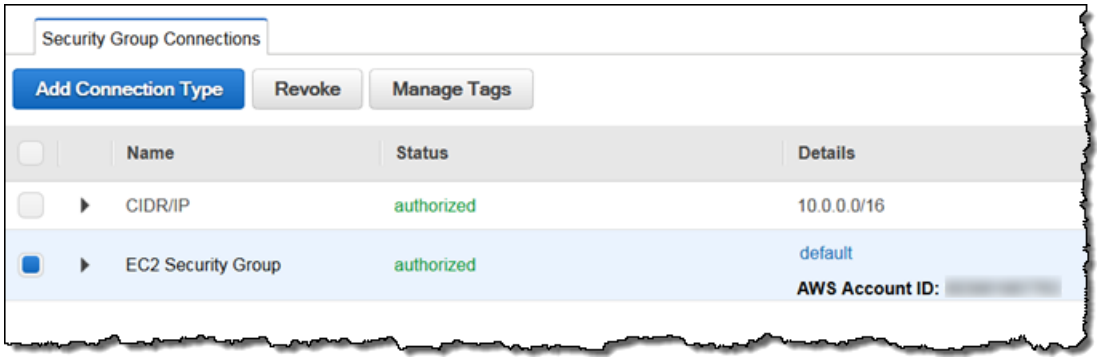
5. Click **Create**.

   The new group will be displayed in the list of cluster security groups.

Tagging a Cluster Security Group

To tag a cluster security group

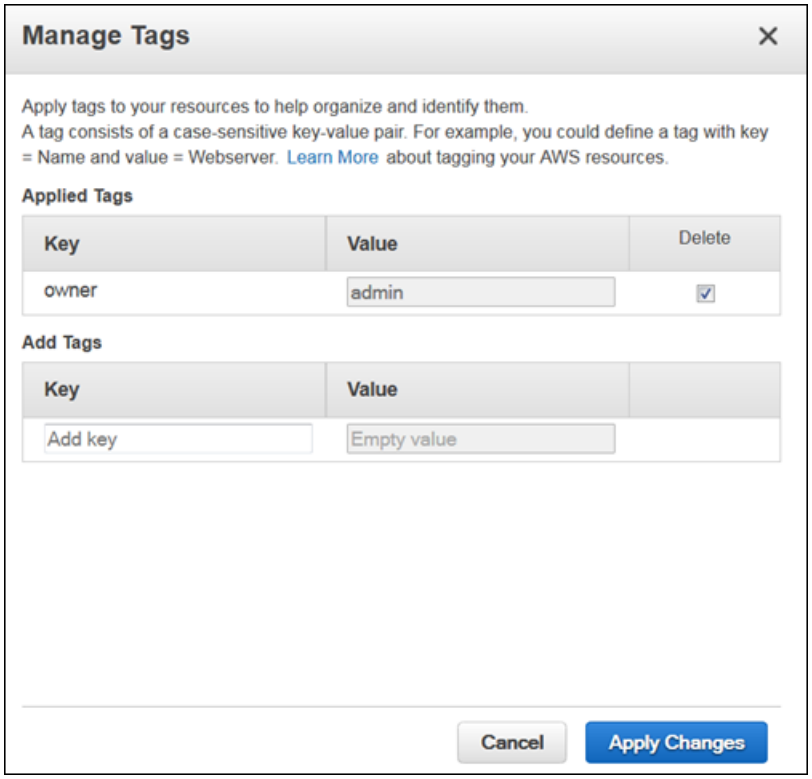
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Security Groups** tab, select the cluster security group and click **Manage Tags**.



4. In the **Manage Tags** dialog box, do one of the following:

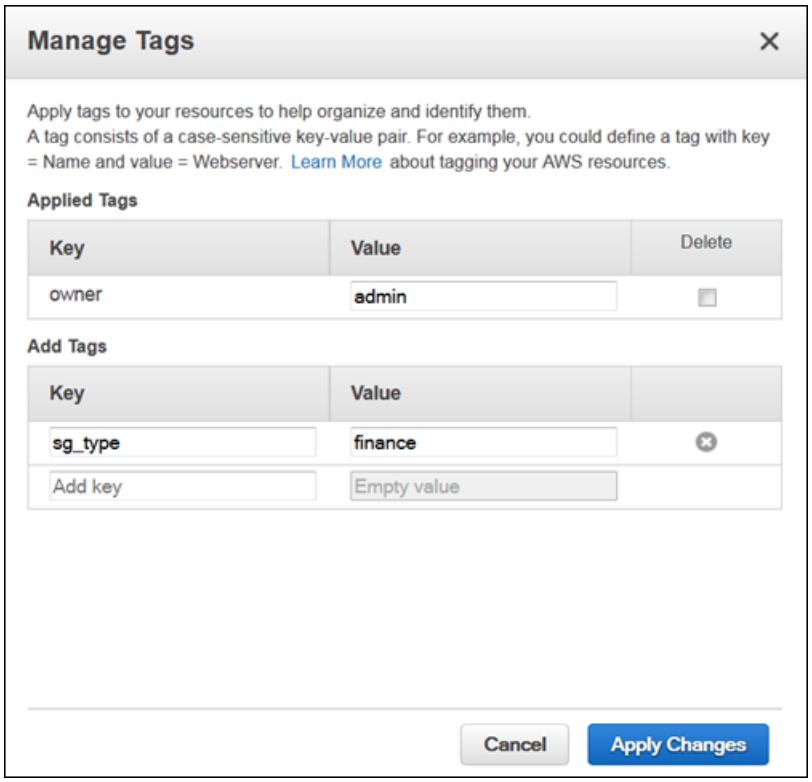
   a. Remove a tag.

   - In the **Applied Tags** section, select **Delete** next to the tag you want to remove.
   - Click **Apply Changes**.



b. Add a tag.

- In the **Add Tags** section, type a key/value pair for the tag.
- Click **Apply Changes**.

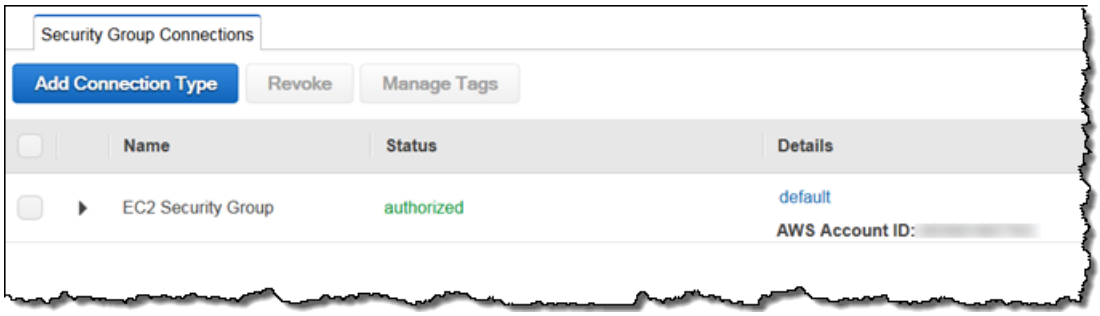


For more information about tagging an Amazon Redshift resource, see How to Manage Tags in the Amazon Redshift Console (p. 243).

Managing Ingress Rules for a Cluster Security Group

To manage ingress rules for a cluster security group

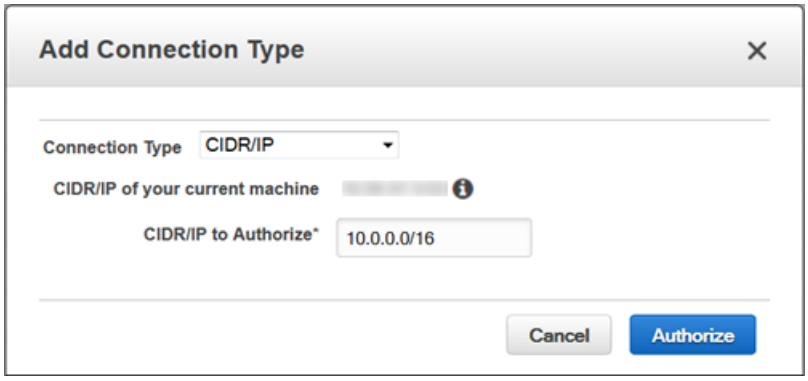
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Security Groups** tab, in the cluster security group list, click the cluster security group whose rules you want to manage.
4. On the **Security Group Connections** tab, click **Add Connection Type**.



5. In the **Add Connection Type** dialog, do one of the following:

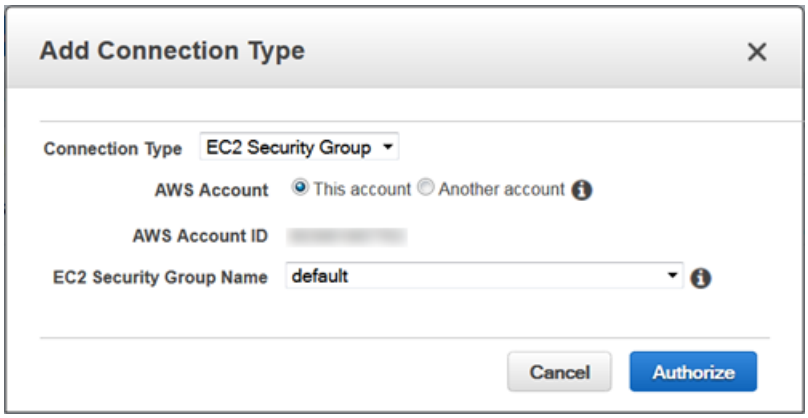
   a. Add an ingress rule based on CIDR/IP.

   - In the **Connection Type** box, click **CIDR/IP**.
   - In the **CIDR/IP to Authorize** box, specify the range.
   - Click **Authorize**.



   b. Add an ingress rule based on an EC2 Security Group.

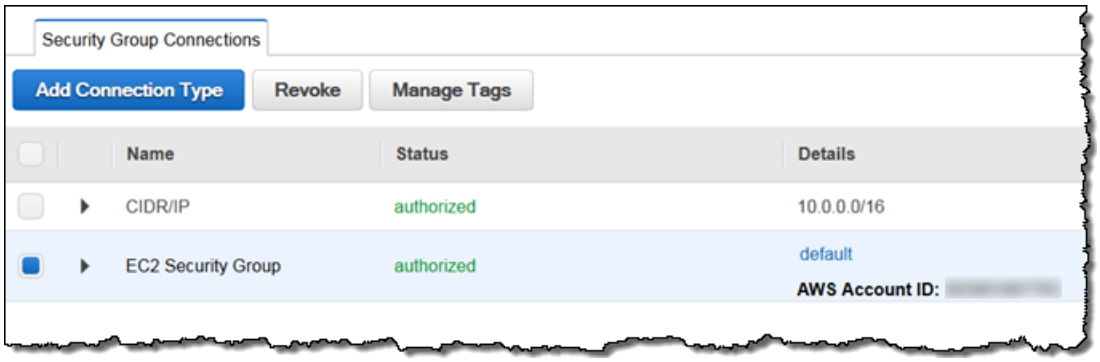
   - Under **Connection Type**, select **EC2 Security Group**.
   - Select the AWS account to use. By default, the account currently logged into the console is used. If you select **Another account**, you must specify the AWS account ID.
   - Click the name of the EC2 security group you want in the **EC2 Security Group Name** box.
   - Click **Authorize**.



Revoking Ingress Rules for a Cluster Security Group

To revoke ingress rules for a cluster security group

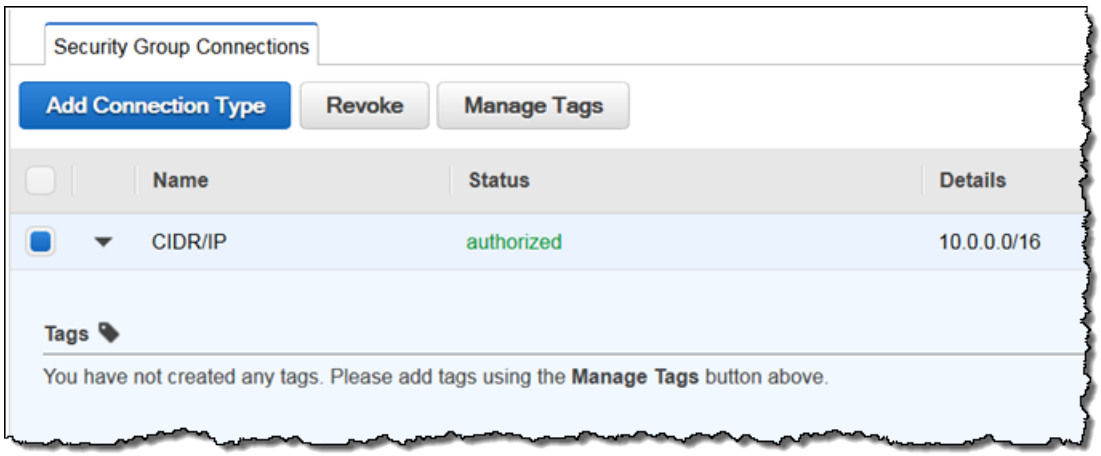
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Security Groups** tab, in the cluster security group list, click the cluster security group whose rules you want to manage.
4. On the **Security Group Connections** tab, select the rule you want to remove and click **Revoke**.



Tagging Ingress Rules for a Cluster Security Group

To tag ingress rules for a cluster security group

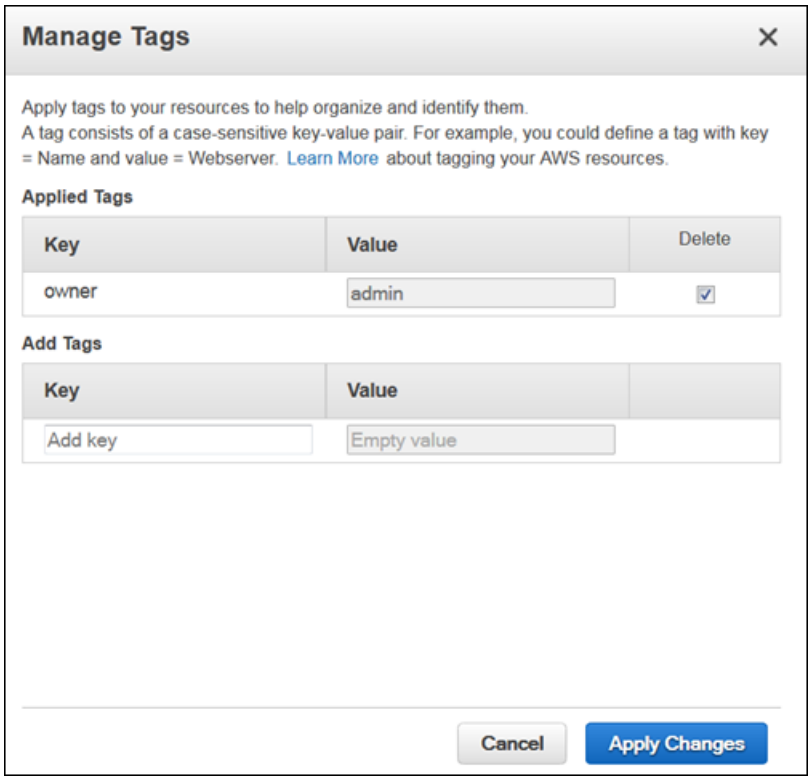
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Security Groups** tab, click the cluster security group whose rules you want to manage.
4. On the **Security Group Connections** tab, select the rule you want to tag and click **Manage Tags**.



5. In the **Manage Tags** dialog box, do one of the following:

a. Remove a tag.

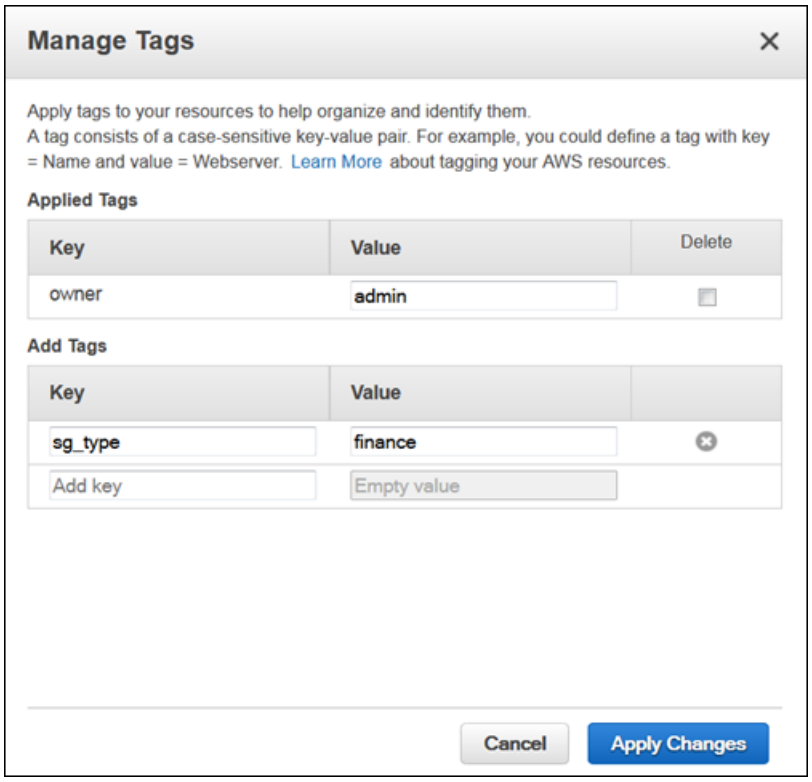
- In the **Applied Tags** section, select **Delete** next to the tag you want to remove.
- Click **Apply Changes**.



b. Add a tag.

Note
Tagging an EC2 Security Group rule only tags that rule, not the EC2 Security Group itself. If you want the EC2 Security Group tagged as well, you must do that separately.

- In the **Add Tags** section, type a key/value pair for the tag.
- Click **Apply Changes**.



For more information about tagging an Amazon Redshift resource, see How to Manage Tags in the Amazon Redshift Console (p. 243).

Deleting a Cluster Security Group

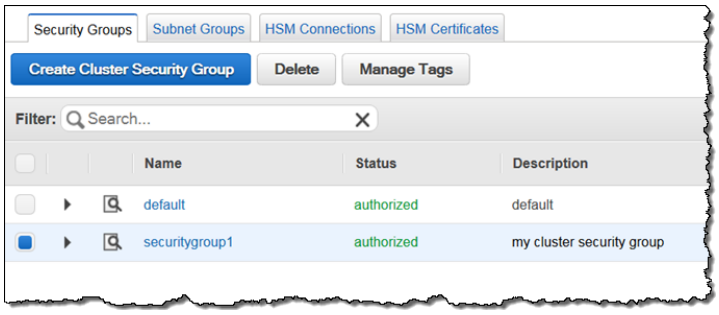
If a cluster security group is associated with one or more clusters, you cannot delete it.

To delete a cluster security group

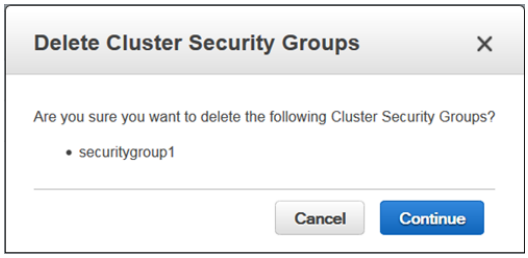
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**.
3. On the **Security Groups** tab, select the cluster security group that you want to delete, and then click **Delete**.

   One row must be selected for the **Delete** button to be enabled.

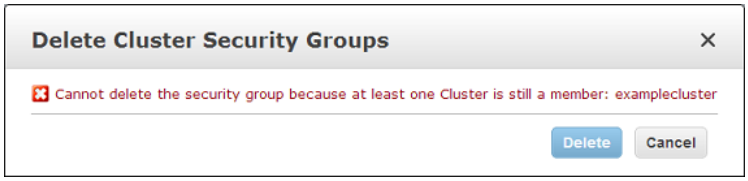
   Note
   You cannot delete the `default` cluster security group.



4. In the **Delete Cluster Security Groups** dialog box, click **Continue**.


Delete Cluster Security Groups
Are you sure you want to delete the following Cluster Security Groups?
• securitygroup1
Cancel
Continue

If the cluster security group is used by a cluster, you will not be able to delete it. The following example shows that `securitygroup1` is used by `examplecluster2`.


Delete Cluster Security Groups
Cannot delete the security group because at least one Cluster is still a member: examplecluster
Delete
Cancel

Associating a Cluster Security Group with a Cluster

Each cluster you provision on the EC2-Classic platform has one or more cluster security groups associated with it. You can associate a cluster security group with a cluster when you create the cluster, or you can associate a cluster security group later by modifying the cluster. For more information, see To create a cluster (p. 14) and To modify a cluster (p. 21). If you are on the EC2-VPC platform, see Managing VPC Security Groups for a Cluster (p. 35) for more information about associating VPC security groups with your cluster.

Managing Cluster Security Groups Using the AWS SDK for Java

The following example demonstrates common operations on cluster security groups, including:

- Creating a new cluster security group.
- Adding ingress rules to a cluster security group.
- Associating a cluster security group with a cluster by modifying the cluster configuration.

By default, when a new cluster security group is created, it has no ingress rules. This example modifies a new cluster security group by adding two ingress rules. One ingress rule is added by specifying a CIDR/IP range; the other is added by specifying an owner ID and Amazon EC2 security group combination.

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and provide a cluster identifier and AWS account number.

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterSecurityGroup {

    public static AmazonRedshiftClient client;
    public static String clusterSecurityGroupName = "securitygroup1";
   public static String clusterIdentifier = "***provide cluster identifier***";

    public static String ownerID = "***provide account id****";

    public static void main(String[] args) throws IOException {

        AWSCredentials credentials = new PropertiesCredentials(
                CreateAndModifyClusterSecurityGroup.class
                        .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             createClusterSecurityGroup();
             describeClusterSecurityGroups();
             addIngressRules();
             associateSecurityGroupWithCluster();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createClusterSecurityGroup() {
        CreateClusterSecurityGroupRequest request = new CreateClusterSecurity
GroupRequest()
        .withDescription("my cluster security group")
        .withClusterSecurityGroupName(clusterSecurityGroupName);

        client.createClusterSecurityGroup(request);
        System.out.format("Created cluster security group: '%s'\n", clusterSe
curityGroupName);
    }

    private static void addIngressRules() {

        AuthorizeClusterSecurityGroupIngressRequest request = new Authorize
ClusterSecurityGroupIngressRequest()
```

```
            .withClusterSecurityGroupName(clusterSecurityGroupName)
            .withCIDRIP("192.168.40.5/32");

        ClusterSecurityGroup result = client.authorizeClusterSecurityGroupIn
gress(request);

        request = new AuthorizeClusterSecurityGroupIngressRequest()
            .withClusterSecurityGroupName(clusterSecurityGroupName)
            .withEC2SecurityGroupName("default")
            .withEC2SecurityGroupOwnerId(ownerID);
        result = client.authorizeClusterSecurityGroupIngress(request);
        System.out.format("\nAdded ingress rules to security group '%s'\n",
clusterSecurityGroupName);
        printResultSecurityGroup(result);
    }

    private static void associateSecurityGroupWithCluster() {

        // Get existing security groups used by the cluster.
        DescribeClustersRequest request = new DescribeClustersRequest()
        .withClusterIdentifier(clusterIdentifier);

        DescribeClustersResult result = client.describeClusters(request);
        List<ClusterSecurityGroupMembership> membershipList =
            result.getClusters().get(0).getClusterSecurityGroups();

        List<String> secGroupNames = new ArrayList<String>();
        for (ClusterSecurityGroupMembership mem : membershipList) {
            secGroupNames.add(mem.getClusterSecurityGroupName());
        }
        // Add new security group to the list.
        secGroupNames.add(clusterSecurityGroupName);

        // Apply the change to the cluster.
        ModifyClusterRequest request2 = new ModifyClusterRequest()
        .withClusterIdentifier(clusterIdentifier)
        .withClusterSecurityGroups(secGroupNames);

        Cluster result2 = client.modifyCluster(request2);
        System.out.format("\nAssociated security group '%s' to cluster '%s'.",
 clusterSecurityGroupName, clusterIdentifier);
   }

    private static void describeClusterSecurityGroups() {
       DescribeClusterSecurityGroupsRequest request = new DescribeClusterSecur
ityGroupsRequest();

        DescribeClusterSecurityGroupsResult result = client.describeClusterSe
curityGroups(request);
        printResultSecurityGroups(result.getClusterSecurityGroups());
    }

    private static void printResultSecurityGroups(List<ClusterSecurityGroup>
groups)
    {
        if (groups == null)
        {
            System.out.println("\nDescribe cluster security groups result is
```

```
null.");
            return;
        }

        System.out.println("\nPrinting security group results:");
        for (ClusterSecurityGroup group : groups)
        {
            printResultSecurityGroup(group);
        }
    }
    private static void printResultSecurityGroup(ClusterSecurityGroup group) {

        System.out.format("\nName: '%s', Description: '%s'\n",
group.getClusterSecurityGroupName(), group.getDescription());
        for (EC2SecurityGroup g : group.getEC2SecurityGroups()) {
           System.out.format("EC2group: '%s', '%s', '%s'\n", g.getEC2Security
GroupName(), g.getEC2SecurityGroupOwnerId(), g.getStatus());
        }
        for (IPRange range : group.getIPRanges()) {
            System.out.format("IPRanges: '%s', '%s'\n", range.getCIDRIP(),
range.getStatus());

        }
    }
}
```

Manage Cluster Security Groups Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage cluster security groups.

- authorize-cluster-security-group-ingress
- create-cluster-security-group
- delete-cluster-security-group
- describe-cluster-security-groups
- revoke-cluster-security-group-ingress

You can use the following Amazon Redshift APIs to manage cluster security groups.

- AuthorizeClusterSecurityGroupIngress
- CreateClusterSecurityGroup
- DeleteClusterSecurityGroup
- DescribeClusterSecurityGroups
- RevokeClusterSecurityGroupIngress

Amazon Redshift Parameter Groups

Topics

- Overview (p. 56)
- Configuring the Maximum Size of a Cursor Result Set (p. 57)
- Configuring Workload Management (p. 59)
- Managing Parameter Groups Using the Console (p. 72)
- Managing Parameter Groups Using the AWS SDK for Java (p. 77)
- Managing Parameter Groups Using the Amazon Redshift CLI and API (p. 80)

Overview

When you provision an Amazon Redshift cluster, you associate a parameter group with it. The parameters in the group apply to all the databases you create on the cluster. The Amazon Redshift engine provides a default parameter group, which has preset values for the parameters. You cannot change the values in the default parameter group. The default parameter group is named `default.redshift-`*`version`*. For example, `default.redshift-1.0`.

The list of parameters available in a parameter group depends on the parameter group family to which it belongs. A parameter group family refers to the Amazon Redshift engine version. For example, parameter group family "redshift-1.0" identifies Amazon Redshift engine version 1.0. Parameter groups in this family have a specific set of parameters for a specific Amazon Redshift version. The following table lists the parameters and their values in the `default` parameter group in the redshift-1.0 parameter group family.

> **Note**
> At this time, version 1.0 of the Amazon Redshift engine is the only version, and so redshift-1.0 is the only parameter group family. The `default` parameter group and any parameter groups that you create belong to the family redshift-1.0.

Parameter Name	Value
datestyle	ISO, MDY
enable_user_activity_logging	false

Parameter Name	Value
extra_float_digits	0
max_cursor_result_set_size	default
query_group	default
require_ssl	false
search_path	$user, public
statement_timeout	0
wlm_json_configuration	[{"query_concurrency":5}]

Use the following topics for more information about these database parameters:

- *enable_user_activity_logging* parameter: Database Audit Logging (p. 215).
- *max_cursor_result_set_size* parameter: Configuring the Maximum Size of a Cursor Result Set (p. 57)
- *require_ssl* parameter: Configure Security Options for Connections (p. 158).
- *wlm_json_configuration* parameter: Configuring Workload Management (p. 59).
- For the other database parameters: Configuration Reference in the *Amazon Redshift Database Developer Guide*.

If your application requires different parameter values, you can create your own parameter group. Creating parameter group is independent of creating clusters. You can create a parameter group before you create a cluster. You associate a parameter group with a cluster when you create it, or you can accept the `default` parameter group.

When you create a parameter group, it will initially have the same parameter values as the `default` parameter group; however, you can modify the parameter values.

In the `default` parameter group or a new parameter group, all parameters are of the `engine-default` type, because the parameter values are preset by Amazon Redshift. When you change a parameter value in a parameter group that you created, the parameter type changes to `user` to indicate that the value has been modified.

For parameter groups that you create, you can modify a parameter value at any time, or you can reset all parameter values to their defaults. You can also associate a different parameter group with a cluster. If you modify parameter values in a parameter group that is already associated with a cluster or you associate a different parameter group with the cluster, you must restart the cluster, or the database for the updated parameter values to take effect. If the cluster fails and is restarted by Amazon Redshift, your changes will be applied at that time.

Configuring the Maximum Size of a Cursor Result Set

The *max_cursor_result_set_size* parameter defines the maximum result set size, in megabytes, that can be stored per individual cursor. This parameter value also affects the number of concurrently active cursors for the cluster, which enables you to configure a value that increases or decreases the

number of cursors for your cluster, and to achieve the ideal balance between maximum result set size and concurrency.

Important
Before you modify this parameter, you should understand the considerations involved with using cursors and the impact that they have on query performance. For more information about the considerations, see Cursor Constraints in the Amazon Redshift Database Developer Guide.

Each Amazon Redshift cluster has a maximum size of data that can be stored across all currently active cursors. This limit is not configurable and is different for each node type. The number of concurrent cursors that you can have for your cluster is determined by the maximum total size allowed for your node type divided by the max-cursor-result-set-size parameter value.

For example, if you have a DW1 XL single node cluster, and you use the default value of two cursors and 32000 MB of data per cursor, the ratio in megabytes is 64000/32000=2. This means that you can have up to two concurrent active cursors, each storing 32000 megabytes of data. However, if you want to have four concurrent cursors for this same node type, you can set *max_cursor_result_set_size* to 16000. The ratio then becomes 64000/16000=4.

The number of concurrent cursors cannot exceed the maximum number of concurrent queries. Amazon Redshift limits the concurrent queries at 50, and adjusts the *max_cursor_result_set_size* accordingly if the value you set causes the concurrent cursor number to be higher.

The following table shows the maximum total result set size for each cluster node type, the default maximum cursor result set, and the number of concurrent cursors based on the default maximum cursor size. Result set sizes are in megabytes:

Node type	Maximum result set per cluster (MB)	Maximum result set per cursor (default)	Concurrent cursors (default)
DW1 XL single node	64000	32000	2
DW1 XL multiple nodes	1800000	450000	4
DW1 8XL multiple nodes	14400000	960000	15
DW2 Large single node	16000	16000	1
DW2 Large multiple nodes	384000	192000	2
DW2 8XL multiple nodes	3000000	750000	4

When you create a parameter group, you can set any value in the globally allowed range for *max_cursor_result_set_size*. However, once the parameter group is associated with a cluster, Amazon Redshift performs validation to ensure that the *max_cursor_result_set_size* parameter is within the allowable range. This validation occurs when you create, modify, or resize an existing cluster and when you modify an existing parameter group that is already associated with one or more clusters. If the parameter group is associated with more than one cluster, Amazon Redshift uses the lowest value allowed for all of the clusters. If the value is not valid, an error is returned that indicates that the parameter group is incompatible with the cluster because the *max_cursor_result_set_size* parameter is out

of range for the cluster. If you change the parameter value and want to return to the default setting, reset the parameter to `0`.

After you change this value, and restart the cluster, you can query STV_CURSOR_CONFIGURATION to make sure that the values took effect.

When running queries, if you get the following error, consider decreasing the *`max_cursor_result_set_size`* value:

"Exceeded the maximum number of cursors allowed to be open concurrently. Analyze the current state with stv_active_cursors and the current configuration via stv_cursor_configuration, and consider decreasing the value of the max_cursor_result_set_size configuration parameter to increase the degree of parallelism."

Conversely, if you get the following error when running queries, consider increasing the *`max_cursor_result_set_size`* value:

"Exceeded the maximum size allowed for the result set of a cursor operation. Current size: *size*. Analyze the current configuration via stv_cursor_configuration, and consider increasing the value of the max_cursor_result_set_size configuration parameter."

Configuring Workload Management

In Amazon Redshift, you use workload management (WLM) to define the number of query queues that are available, and how queries are routed to those queues for processing. WLM is part of parameter group configuration. A cluster uses the WLM configuration that is specified in its associated parameter group.

You can configure WLM by using the Amazon Redshift console, the AWS CLI, Amazon Redshift API, or one of the AWS SDKs. WLM configuration comprises several properties to define queue behavior, such as memory allocation across queues, the number of queries that can run concurrently in a queue, and so on. The following list describes the WLM properties that you can configure for each queue.

Console Field	JSON Property	Value
Concurrency	*query_concurrency*	The number of queries that can run concurrently in a queue. Amazon Redshift will allocate this amount of memory to the query and the queue will execute with this many slots. Valid values are integers 1–50.

Console Field	JSON Property	Value
User Groups	*user_group*	A comma-separated list of user group names. When members of one of these user groups run queries in the database, their queries are routed to queues associated with the WLM configuration. For more information, see Assigning Queries to Queues.

Console Field	JSON Property	Value
User Group Wildcard	*user_group_wild_card*	

Console Field	JSON Property	Value
		A re lo B eu la v ta ht - n i - c id se t a - ht e w r e o t - ne el ba - dl iw sd ra c ro f re su . sp o g f I si ht s i , 0 - dl iw sd ra c e r a - si d , el la f i si ht s i , 1 - dl iw sd ra c e r a - ne . el la ne hW - dl iw sd ra c e r a - ne , el la uo y na c e s u " * " r o " ? " o t - ep s yf ic - l u n el pi t re su sp o g ne hw

Console Field	JSON Property	eulaV
		-nur gnin -req .sei
Query Groups	*query_group*	A -aoc -pes - r a detail f o yreuq .spog nehW -mem sreb f o eht yreuq puorg nu r -req se i n i eht -atad ,essb riet -req se i e r a dettor o t eht euuq tant s i - s a - o s - i c detail ht iw riet yreuq .pog

Console Field	JSON Property	Value
Query Group Wildcard	*query_group_wild_card*	

Console Field	JSON Property	Value
		A re lo B eu la v ta ht - n i - c id se t a - ht ew r e o t - ne el ba - dl iw sd ra c r o f yr eu q . sp og f I si ht s i , 0 - dl iw sd ra c e r a - si d , el la f i si ht s i , 1 - dl iw sd ra c e r a - ne . el la ne hW - dl iw sd ra c e r a - ne , el la uo y na c es u " * " r o " ? " o t - ep s yf ic - l un el pi t yr eu q sp or g ne hw

Console Field	JSON Property	eulaV
		-nur gnin -req .sei
Timeout	*max_execution_time*	ehT -xam -mi m u ,sei t n i -lim -il ,sdes -req sei nac nur -eb erof -eb gni -nac edk e

Console Field	JSON Property	eulaV
% Memory	*memory_percent_to_use*	eht tnuoma fo yromem ot -lacol eta o teh .eueq eht lacot tnuoma fo yromem sorca la seueq tsum laup 001 -rep ,tnec o sfi uoy tes siht -pord yterf eno ,eueq uoy tsum tes eht -hto re seueq -ca -drocyb gi

For more information about each of these properties, go to Defining Query Queues in the *Amazon Redshift Database Developer Guide*.

When you create a parameter group, the *default WLM configuration* is one queue that can run up to five queries concurrently. You can add additional queues and configure WLM properties in each of them if you want more control over query processing. Each queue that you add has the same default WLM configuration until you configure its properties.

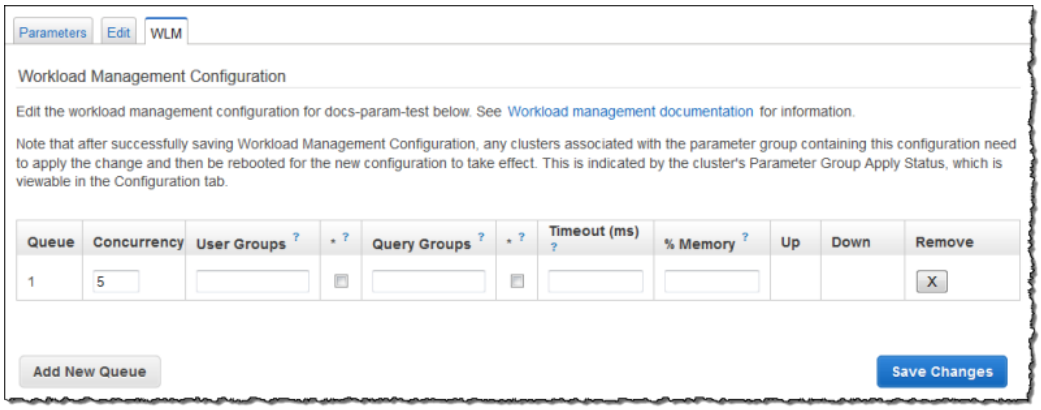
The last queue specified in the WLM configuration is the *default queue*. Unless a query is routed to another queue based on criteria in the WLM configuration, it is processed by the default queue. You cannot specify

user groups or query groups for the default queue. The default queue must always be the last queue in the WLM configuration.

As with other parameters, you cannot modify the WLM configuration in the default parameter group. Clusters associated with the default parameter group always use the default WLM configuration. If you want to modify the WLM configuration, you must create a parameter group and then associate that parameter group with any clusters that require your custom WLM configuration.

Configuring the wlm_json_configuration Parameter

To configure WLM, you modify the *wlm_json_configuration* parameter and format it in JavaScript Object Notation (JSON). If you configure WLM by using the Amazon Redshift console, you do not need to understand JSON formatting because the console provides an easy way to add queues and configure their properties. For more information about configuring WLM by using the Amazon Redshift console, see Modifying a Parameter Group (p. 74). The following screenshot shows the WLM configuration in the Amazon Redshift console.


Parameters
Edit
WLM
Workload Management Configuration
Edit the workload management configuration for docs-param-test below. See Workload management documentation for information.
Note that after successfully saving Workload Management Configuration, any clusters associated with the parameter group containing this configuration need to apply the change and then be rebooted for the new configuration to take effect. This is indicated by the cluster's Parameter Group Apply Status, which is viewable in the Configuration tab.
Queue
Concurrency
User Groups
*
Query Groups
*
Timeout (ms)
% Memory
Up
Down
Remove
1
5
X
Add New Queue
Save Changes

If you configure WLM by using the AWS CLI, Amazon Redshift API, or one of the AWS SDKs, use the rest of this section to learn how to construct the JSON structure for the *wlm_json_configuration* parameter. The following example is the default WLM configuration, which defines one queue with a concurrency level of five:

```
{
   "query_concurrency":5
}
```

The default WLM configuration is very simple, with only queue and one property. You can add more queues and configure multiple properties for each in the JSON structure. The following syntax represents the JSON structure that you use to configure multiple queues with multiple properties:

```
[
   {
      "ParameterName":"wlm_json_configuration", "ParameterValue":
         "[
             {
                "q1_first_property_name":"q1_first_property_value",
                "q1_second_property_name":"q1_second_property_value",
                ...
             },
             {
                "q2_first_property_name":"q2_first_property_value",
```

```
                "q2_second_property_name":"q2_second_property_value",
                ...
            }
            ...

        ]"
    }
]
```

In the preceding example, the representative properties that begin with **q1** are objects in an array for the first queue. Each of these objects is a name/value pair; `name` and `value` together set the WLM properties for the first queue. The representative properties that begin with **q2** are objects in an array for the second queue. If you require more queues, you add another array for each additional queue and set the properties for each object.

Configuring WLM by Using the AWS CLI

To configure Amazon Redshift parameters by using the AWS CLI, you use the `modify-cluster-parameter-group` command for a specific parameter group. You specify the parameter group to modify in *`parameter-group-name`*. You use the *`parameters`* parameter (for the `modify-cluster-parameter-group` command) to specify name/value pairs for each parameter that you want to modify in the parameter group. Separate each parameter from others with a space if you modify multiple parameters in the same command.

The *`parameters`* parameter (for the `modify-cluster-parameter-group` command) uses the following syntax, where *`parameter_name`* and *`parameter_value`* are each replaced with an actual parameter to modify and a value for that parameter.

```
--parameters ParameterName=parameter_name,ParameterValue=parameter_value
```

The following example shows how to configure the *`statement_timeout`* and *`enable_user_activity_logging`* parameters for the `myclusterparametergroup` parameter group.

Note
For readability purposes, the example is displayed on two lines, but in the actual AWS CLI this would be one line.

```
aws redshift modify-cluster-parameter-group --parameter-group-name mycluster
parametergroup
--parameters ParameterName=statement_timeout,ParameterValue=20000 Parameter
Name=enable_user_activity_logging,ParameterValue=true
```

The *`wlm_json_configuration`* parameter requires a specific format when you use the AWS CLI. The format that you use depends on your client operating system. Operating systems have different ways to enclose the JSON structure so it's passed correctly from the command line. For details on how to construct the appropriate command in the Linux, Mac OS X, and Windows operating systems, see the sections following.

For more information about the differences in enclosing JSON data structures in the AWS CLI in general, see Quoting Strings in the *AWS Command Line Interface User Guide*. For more information about workload management, go to Implementing Workload Management in the *Amazon Redshift Developer Guide*.

Rules for Configuring WLM by Using the AWS CLI in the Command Line on the Linux and Mac OS X Operating Systems

- The entire JSON structure must be enclosed in single quotation marks (') and brackets ([ ]).
- All parameter names and parameter values must be enclosed in double quotation marks (").
- Within the *ParameterValue* value, you must enclose the entire nested structure in double-quotation marks (") and brackets ([ ]).
- Within the nested structure, each of the properties and values for each queue must be enclosed in curly braces ({ }).
- Within the nested structure, you must use the backslash (\) escape character before each double-quotation mark (").
- For name/value pairs, a colon (:) separates each property from its value.
- Each name/value pair is separated from another by a comma (,).
- Multiple queues are separated by a comma (,) between the end of one queue's curly brace (}) and the beginning of the next queue's curly brace ({).

The following example shows how to configure two queues with different properties for a cluster parameter group named `myclusterparametergroup`. This example works for the Linux and Mac OS X operating systems.

Note
For readability purposes, the example is displayed on two lines, but in the actual AWS CLI this would be one line.

```
aws redshift modify-cluster-parameter-group --parameter-group-name mycluster
parametergroup
--parameters '[{"ParameterName":"wlm_json_configuration", "Parameter
Value":"[{\"query_group\":[\"test\"],\"concurrency_level\":7},{\"concur
rency_level\":5}]"}]'
```

Rules for Configuring WLM by Using the AWS CLI in Windows PowerShell on Microsoft Windows Operating Systems

- The entire JSON structure must be enclosed in single quotation marks (') and brackets ([ ]).
- All parameter names and parameter values must be enclosed in double quotation marks (").
- Within the *ParameterValue* value, you must enclose the entire nested structure in double-quotation marks (") and brackets ([ ]).
- Within the nested structure, each of the properties and values for each queue must be enclosed in curly braces ({ }).
- Within the nested structure, you must use the backslash (\) escape character before each double-quotation mark (") and its backslash (\) escape character. This requirement means that you will use three backslashes and a double quotation mark to make sure that the properties are passed in correctly (\\\:).
- For name/value pairs, a colon (:) separates each property from its value.
- Each name/value pair is separated from another by a comma (,).
- Multiple queues are separated by a comma (,) between the end of one queue's curly brace (}) and the beginning of the next queue's curly brace ({).

The following example shows how to configure two queues with different properties for a cluster parameter group named `myclusterparametergroup`. This example works for Windows PowerShell on Windows operating systems.

Note
For readability purposes, the example is displayed on two lines, but in the actual AWS CLI this would be one line.

```
aws redshift modify-cluster-parameter-group --parameter-group-name mycluster
parametergroup
--parameters '[{\"ParameterName\":\"wlm_json_configuration\", \"Parameter
Value\":\"[{\\\"query_group\\\":[\\\"test\\\"],\\\"concur
rency_level\\\":7},{\\\"concurrency_level\\\":5}]\"}]'
```

Rules for Configuring WLM by Using the Command Prompt on Windows Operating Systems

- The entire JSON structure must be enclosed in double-quotation marks (") and brackets ([ ]).
- All parameter names and parameter values must be enclosed in double quotation marks (").
- Within the *`ParameterValue`* value, you must enclose the entire nested structure in double-quotation marks (") and brackets ([ ]).
- Within the nested structure, each of the properties and values for each queue must be enclosed in curly braces ({ }).
- Within the nested structure, you must use the backslash (\) escape character before each double-quotation mark (") and its backslash (\) escape character. This requirement means that you will use three backslashes and a double quotation mark to make sure that the properties are passed in correctly (\\\:).
- For name/value pairs, a colon (:) separates each property from its value.
- Each name/value pair is separated from another by a comma (,).
- Multiple queues are separated by a comma (,) between the end of one queue's curly brace (}) and the beginning of the next queue's curly brace ({).

The following example shows how to configure two queues with different properties for a cluster parameter group named `myclusterparametergroup`. This example works when using the command prompt on Windows operating systems.

Note
For readability purposes, the example is displayed on two lines, but in the actual AWS CLI this would be one line.

```
aws redshift modify-cluster-parameter-group --parameter-group-name mycluster
parametergroup
--parameters "[{\"ParameterName\":\"wlm_json_configuration\", \"Parameter
Value\":\"[{\\\"query_group\\\":[\\\"test\\\"],\\\"concur
rency_level\\\":7},{\\\"concurrency_level\\\":5}]\"}]"
```

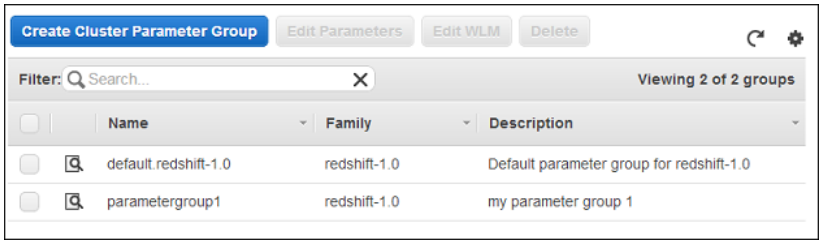
Managing Parameter Groups Using the Console

Topics

- Creating a Parameter Group (p. 73)
- Modifying a Parameter Group (p. 74)
- Deleting a Parameter Group (p. 76)
- Associating a Parameter Group with a Cluster (p. 77)

You can create, modify, and delete parameter groups by using the Amazon Redshift console. All of the tasks start from the parameter group list.

In the parameter group list below, there are two parameter groups, the default parameter group and a custom parameter group called `parametergroup1`. Because `parametergroup1` is selected (highlighted), you can edit or delete it.



You cannot delete or modify the default parameter group. If you want a parameter group that has different parameter values, you must create a new parameter group and modify it. The default parameter group is not created until you have launched a cluster.

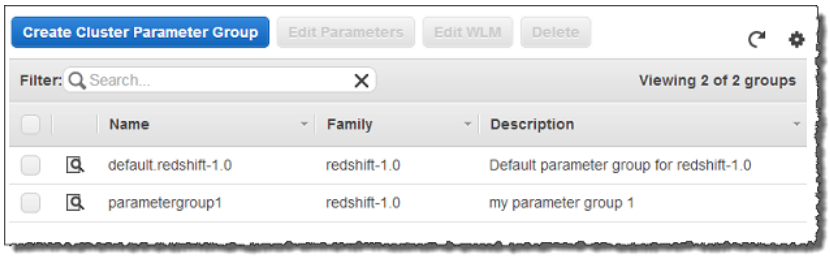
Each cluster must be associated with one parameter group. You can make this association when you create a cluster, or you can modify an existing cluster to use a different parameter group. For more information about associating a parameter group with a cluster, see To modify a cluster (p. 21). When you change the settings in a parameter group that is associated with a cluster, the change is not applied until the cluster is rebooted.

Creating a Parameter Group

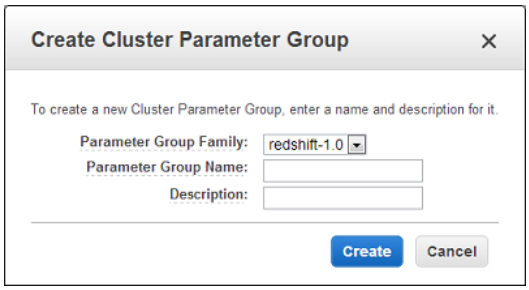
You can create a parameter group if you want to set parameter values that are different from the default parameter group for your cluster version.

To create a parameter group

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Parameter Groups**.
3. Click **Create Cluster Parameter Group**.



4. In the **Create Cluster Parameter Group** dialog box, specify a parameter group family, a parameter group name, and a parameter group description. When the settings are as you want them, click **Yes, Create**.



Modifying a Parameter Group

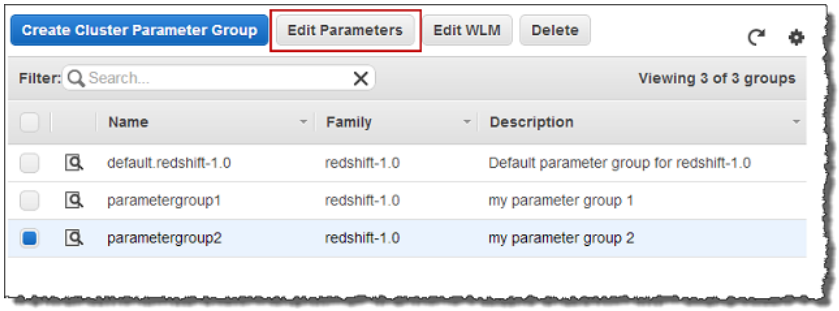
If the parameter group you are modifying is associated with a cluster, then the changes you make to the parameter group apply only after you reboot the cluster.

To modify a parameter group

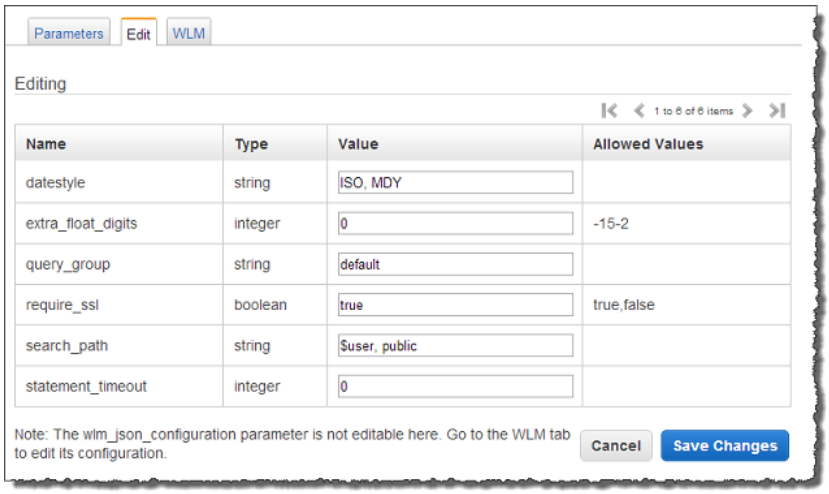
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Parameter Groups**.
3. In the parameter group list, click the row of the parameter group you want to modify, and then click **Edit Parameters**.

   One row must be highlighted for the **Edit Parameters** button to be enabled.

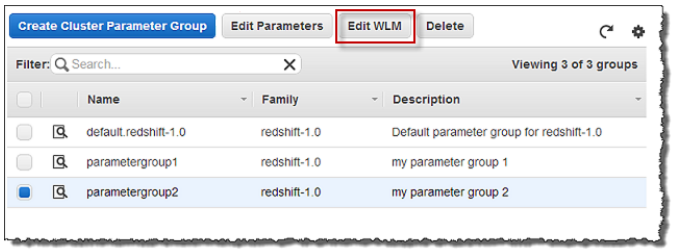
   Note
   You cannot modify the default parameter group.



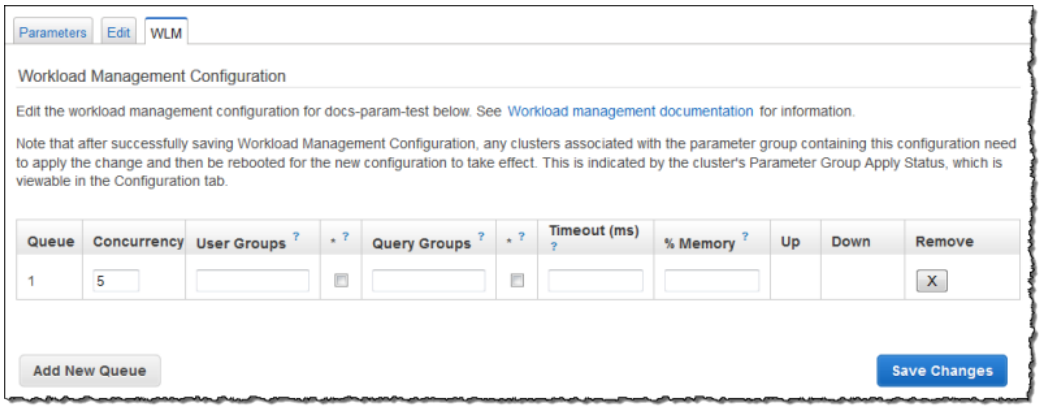
4. Click the **Edit** tab. To modify a parameter value, type a new value in the corresponding box in the **Value** column. When the settings are as you want them, click **Save Changes**.



5. To edit the workload management configuration, in the parameter group list, click the row of the parameter group you want to modify, and then click **Edit WLM**.



Click the **WLM** tab to edit the workload management configuration.


Parameters
Edit
WLM
Workload Management Configuration
Edit the workload management configuration for docs-param-test below. See Workload management documentation for information.
Note that after successfully saving Workload Management Configuration, any clusters associated with the parameter group containing this configuration need to apply the change and then be rebooted for the new configuration to take effect. This is indicated by the cluster's Parameter Group Apply Status, which is viewable in the Configuration tab.
Queue
Concurrency
User Groups
Query Groups
Timeout (ms)
% Memory
Up
Down
Remove
1
5
X
Add New Queue
Save Changes

a. Do one of the following:

To...	Do this...
Modify an existing queue	Change the queue's **Concurrency**, **User Groups**, or **Query Groups** fields. For more information about these queue fields, go to Implementing Workload Management in the *Amazon Redshift Developer Guide*.
Create a new queue	Click **Add New Queue**.

To...	Do this...
Change the order of a queue in the list	Use the **Up** and **Down** buttons to change the order that queries are assigned to queues based on users and query groups. For more information, go to Defining Query Queues in the *Amazon Redshift Developer Guide*.
Delete a queue	Click the **X** button of the queue to remove.

b. When the settings are as you want them, click **Save Changes**.

Deleting a Parameter Group

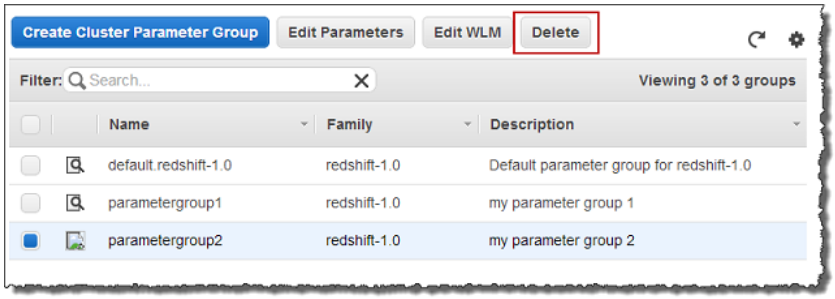
If a parameter group is associated with one or more clusters, you cannot delete it.

To delete a parameter group

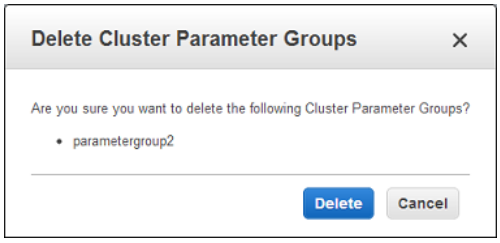
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Parameter Groups**.
3. Click the row of the parameter group that you want to delete, and then click **Delete**.

   One row must be highlighted for the **Delete** button to be enabled.

   Note
   You cannot delete the default parameter group.



4. In the **Delete Cluster Parameter Groups** dialog box, click **Delete**.



Associating a Parameter Group with a Cluster

Each cluster you provision has one parameter group associated with it. You can associate a parameter group with a cluster when you create the cluster, or you can associate a parameter group later by modifying the cluster. For more information, see To create a cluster (p. 14) and To modify a cluster (p. 21).

Managing Parameter Groups Using the AWS SDK for Java

This example demonstrates the following tasks related to parameter groups:

- Creating a parameter group
- Modifying a parameter group
- Associating a parameter group with a cluster
- Getting information about parameter groups

This example creates a new parameter group, `parametergroup1`, and makes the following updates:

- Changes the parameter `extra_float_digits` to 2 from the default value of 0.
- Replaces the existing workload management configuration (`wlm_json_configuration` parameter) with the following JSON which defines a queue in addition to the default queue.

```
[
   {
      "user_group":[
         "example_user_group1"
      ],
      "query_group":[
         "example_query_group1"
      ],
      "query_concurrency":7
   },
   {
      "query_concurrency":5
   }
]
```

The preceding JSON is an array of two objects, one for each queue. The first object defines a queue with specific user group and query group. It also sets the concurrency level to 7.

```
{
   "user_group":[
      "example_user_group1"
   ],
   "query_group":[
      "example_query_group1"
   ],
   "query_concurrency":7
}
```

Because this example replaces the WLM configuration, this JSON configuration also defines the default queue with no specific user group or query group. It sets the concurrency to the default value, 5.

```
{
   "query_concurrency":5
}
```

For more information about Workload Management (WML) configuration, go to Implementing workload management.

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and provide a cluster identifier.

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class CreateAndModifyClusterParameterGroup {

    public static AmazonRedshiftClient client;
    public static String clusterParameterGroupName = "parametergroup1";
   public static String clusterIdentifier = "***provide cluster identifier***";

    public static String parameterGroupFamily = "redshift-1.0";

    public static void main(String[] args) throws IOException {

        AWSCredentials credentials = new PropertiesCredentials(
                CreateAndModifyClusterParameterGroup.class
                        .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             createClusterParameterGroup();
             modifyClusterParameterGroup();
             associateParameterGroupWithCluster();
             describeClusterParameterGroups();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void createClusterParameterGroup() {
       CreateClusterParameterGroupRequest request = new CreateClusterParamet
erGroupRequest()
           .withDescription("my cluster parameter group")
           .withParameterGroupName(clusterParameterGroupName)
           .withParameterGroupFamily(parameterGroupFamily);
       client.createClusterParameterGroup(request);
       System.out.println("Created cluster parameter group.");
    }
```

```
    private static void describeClusterParameterGroups() {
      DescribeClusterParameterGroupsResult result = client.describeClusterPara
meterGroups();
       printResultClusterParameterGroups(result);
    }

    private static void modifyClusterParameterGroup() {
       List<Parameter> parameters = new ArrayList<Parameter>();
       parameters.add(new Parameter()
           .withParameterName("extra_float_digits")
           .withParameterValue("2"));
       // Replace WLM configuration. The new configuration defines a queue (in
 addition to the default).
       parameters.add(new Parameter()
       .withParameterName("wlm_json_configuration")
       .withParameterValue("[{\"user_group\":[\"ex
ample_user_group1\"],\"query_group\":[\"example_query_group1\"],\"query_concur
rency\":7},{\"query_concurrency\":5}]"));

       ModifyClusterParameterGroupRequest request = new ModifyClusterParamet
erGroupRequest()
           .withParameterGroupName(clusterParameterGroupName)
           .withParameters(parameters);
       client.modifyClusterParameterGroup(request);

    }

    private static void associateParameterGroupWithCluster() {

        ModifyClusterRequest request = new ModifyClusterRequest()
        .withClusterIdentifier(clusterIdentifier)
        .withClusterParameterGroupName(clusterParameterGroupName);

        Cluster result = client.modifyCluster(request);

        System.out.format("Parameter Group %s is used for Cluster %s\n",
                clusterParameterGroupName, result.getClusterParamet
erGroups().get(0).getParameterGroupName());
    }
    private static void printResultClusterParameterGroups(DescribeClusterPara
meterGroupsResult result)
    {
        if (result == null)
        {
            System.out.println("\nDescribe cluster parameter groups result is
null.");
            return;
        }

        System.out.println("\nPrinting parameter group results:\n");
        for (ClusterParameterGroup group : result.getParameterGroups()) {
            System.out.format("\nDescription: %s\n", group.getDescription());
          System.out.format("Group Family Name: %s\n", group.getParameterGroup
Family());
          System.out.format("Group Name: %s\n", group.getParameterGroupName());

            describeClusterParameters(group.getParameterGroupName());
```

```
        }
    }

    private static void describeClusterParameters(String parameterGroupName) {

        DescribeClusterParametersRequest request = new DescribeClusterParamet
ersRequest()
            .withParameterGroupName(parameterGroupName);

       DescribeClusterParametersResult result = client.describeClusterParamet
ers(request);
        printResultClusterParameters(result, parameterGroupName);
    }

    private static void printResultClusterParameters(DescribeClusterParamet
ersResult result, String parameterGroupName)
    {
        if (result == null)
        {
            System.out.println("\nCluster parameters is null.");
            return;
        }

       System.out.format("\nPrinting cluster parameters for \"%s\"\n", paramet
erGroupName);
        for (Parameter parameter : result.getParameters()) {
            System.out.println("  Name: " + parameter.getParameterName() + ",
Value: " + parameter.getParameterValue());
            System.out.println("  DataType: " + parameter.getDataType() + ",
MinEngineVersion: " + parameter.getMinimumEngineVersion());
          System.out.println("  AllowedValues: " + parameter.getAllowedValues()
 + ", Source: " + parameter.getSource());
           System.out.println("  IsModifiable: " + parameter.getIsModifiable()
 + ", Description: " + parameter.getDescription());
        }
    }
}
```

Managing Parameter Groups Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage parameter groups.

- create-cluster-parameter-group
- delete-cluster-parameter-group
- describe-cluster-parameters
- describe-cluster-parameter-groups
- describe-default-cluster-parameters
- modify-cluster-parameter-group
- reset-cluster-parameter-group

You can use the following Amazon Redshift APIs to manage parameter groups.

- CreateClusterParameterGroup
- DeleteClusterParameterGroup
- DescribeClusterParameters
- DescribeClusterParameterGroups
- DescribeDefaultClusterParameters
- ModifyClusterParameterGroup
- ResetClusterParameterGroup

Amazon Redshift Snapshots

Topics

- Overview (p. 82)
- Managing Snapshots Using the Console (p. 86)
- Managing Snapshots Using the AWS SDK for Java (p. 94)
- Managing Snapshots Using the Amazon Redshift CLI and API (p. 97)

Overview

Snapshots are point-in-time backups of a cluster. There are two types of snapshots: *automated* and *manual*. Amazon Redshift stores these snapshots internally in Amazon S3 by using an encrypted Secure Sockets Layer (SSL) connection. If you need to restore from a snapshot, Amazon Redshift creates a new cluster and imports data from the snapshot that you specify.

Amazon Redshift periodically takes snapshots and tracks incremental changes to the cluster since the last snapshot. Amazon Redshift retains all of the data required to restore a cluster from a snapshot.

You can monitor the progress of long-running snapshots by either calling describe-cluster-snapshots in the CLI, by calling the DescribeClusterSnapshots API action, or viewing the snapshot details in the AWS Management Console. For an in-progress snapshot, these display information such as the size of the incremental snapshot, the transfer rate, the elapsed time, and the estimated time remaining.

Amazon Redshift provides free storage for snapshots that is equal to the storage capacity of your cluster until you delete the cluster. After you reach the free snapshot storage limit, you are charged for any additional storage at the normal rate. Because of this, you should evaluate how many days you need to keep automated snapshots and configure their retention period accordingly, and delete any manual snapshots that you no longer need. For pricing information, go to the Amazon Redshift product detail page.

Automated Snapshots

When automated snapshots are enabled for a cluster, Amazon Redshift periodically takes snapshots of that cluster. Automated snapshots are enabled by default when you create a cluster. These snapshots are deleted at the end of a retention period. The default retention period is one day, but you can modify it by using the Amazon Redshift console or programmatically by using the Amazon Redshift API.

To disable automated snapshots, set the retention period to zero. If you disable automated snapshots, Amazon Redshift stops taking snapshots and deletes any existing automated snapshots for the cluster.

Only Amazon Redshift can delete an automated snapshot; you cannot delete them manually. Amazon Redshift deletes automated snapshots at the end of a snapshot's retention period, when you disable automated snapshots, or when you delete the cluster. If you want to keep an automated snapshot for a longer period, you can create a copy of it as a manual snapshot. The automated snapshot is retained until the end of retention period, but the corresponding manual snapshot is retained until you manually delete it.

Manual Snapshots

Regardless of whether you enable automated snapshots, you can take a manual snapshot whenever you want. Amazon Redshift will never automatically delete a manual snapshot. Manual snapshots are retained even after you delete your cluster.

Because manual snapshots accrue storage charges, it's important that you manually delete them if you no longer need them. If you delete a manual snapshot, you cannot start any new operations that reference that snapshot. However, if a restore operation is in progress, that restore operation will run to completion.

Copying Snapshots to Another Region

You can configure Amazon Redshift to automatically copy snapshots (automated or manual) for a cluster to another region. When a snapshot is created in the cluster's primary region, it will be copied to a secondary region; these are known respectively as the *source region* and *destination region*. By storing a copy of your snapshots in another region, you have the ability to restore your cluster from recent data if anything affects the primary region. You can configure your cluster to copy snapshots to only one destination region at a time. For a list of Amazon Redshift regions, see Regions and Endpoints in the Amazon Web Services General Reference.

When you enable Amazon Redshift to automatically copy snapshots to another region, you specify the destination region where you want snapshots to be copied. In the case of automated snapshots, you can also specify the retention period that they should be kept in the destination region. After an automated snapshot is copied to the destination region and it reaches the retention time period there, it is deleted from the destination region, keeping your snapshot usage low. You can change this retention period if you need to keep the automated snapshots for a shorter or longer period of time in the destination region.

The retention period that you set for automated snapshots that are copied to the destination region is separate from the retention period for automated snapshots in the source region. The default retention period for copied snapshots is seven days. That seven-day period only applies to automated snapshots. Manual snapshots are not affected by the retention period in either the source or destination regions, and they remain until you manually delete them.

You can disable automatic snapshot copy for a cluster at any time. When you disable this feature, snapshots are no longer copied from the source region to the destination region. Any automated snapshots copied to the destination region are deleted as they reach the retention period limit, unless you create manual snapshot copies of them. These manual snapshots, and any manual snapshots that were copied from the destination region, are retained in the destination region until you manually delete them.

If you want to change the destination region that you copy snapshots to, you have to first disable the automatic copy feature and then re-enable it, specifying the new destination region.

Copying snapshots across regions incurs data transfer charges. Once a snapshot is copied to the destination region, it becomes active and available for restoration purposes.

Restoring a Cluster from a Snapshot

A snapshot contains data from any databases that are running on your cluster, and also information about your cluster, including the number of nodes, node type, and master user name. If you need to restore your cluster from a snapshot, Amazon Redshift uses the cluster information to create a new cluster and then restores all the databases from the snapshot data. The new cluster that Amazon Redshift creates from the snapshot will have same configuration, including the number and type of nodes, as the original cluster from which the snapshot was taken. The cluster is restored in the same region and Availability Zone unless you specify another Availability Zone in your request.

You can monitor the progress of a restore by either calling the DescribeClusters API action, or viewing the cluster details in the AWS Management Console. For an in-progress restore, these display information such as the size of the snapshot data, the transfer rate, the elapsed time, and the estimated time remaining. For a description of these metrics, go to RestoreStatus.

You cannot use a snapshot to revert an active cluster to a previous state.

Note
When you restore a snapshot into a new cluster, the default security group and parameter group are used unless you specify different values.

Sharing Snapshots

You can share an existing manual snapshot with the users in as many as 20 AWS customer accounts by authorizing access to the snapshot. A person logged in as a user in one of the authorized accounts can then describe the snapshot or restore it to create a new Redshift cluster under their account. For example, if you use separate AWS customer accounts for production and test, a user can log on using the production account and share a snapshot with users in the test account. Someone logged on as a test account user can then restore the snapshot to create a new cluster that is owned by the test account for testing or diagnostic work.

A manual snapshot is permanently owned by the AWS customer account under which it was created. Only users in the account owning the snapshot can authorize other accounts to access the snapshot, or to revoke authorizations. Users in the authorized accounts can only describe or restore any snapshot that has been shared with them; they cannot copy or delete snapshots that have been shared with them. An authorization remains in effect until the snapshot owner revokes it. If an authorization is revoked, the previously authorized user loses visibility of the snapshot and cannot launch any new actions referencing the snapshot. If the account is in the process of restoring the snapshot when access is revoked, the restore runs to completion. You cannot delete a snapshot while it has active authorizations; you must first revoke all of the authorizations.

AWS customer accounts are always authorized to access snapshots owned by the account. Attempts to authorize or revoke access to the owner account will receive an error. You cannot restore or describe a snapshot that is owned by an inactive AWS customer account.

After you have authorized access to an AWS customer account, no IAM users in that account can perform any actions on the snapshot unless they have IAM policies that allow them to do so.

- IAM users in the snapshot owner account can authorize and revoke access to a snapshot only if they have an IAM policy that allows them to perform those actions with a resource specification that includes the snapshot. For example, the following policy allows a user in AWS account `012345678912` to authorize other accounts to access a snapshot named `my-snapshot20130829`:

```
{
  "Version": "2012-10-17",
  "Statement":[
    {
```

```
     "Effect":"Allow",
     "Action":[
         "redshift:AuthorizeSnapshotAccess",
         "redshift:RevokeSnapshotAccess"
         ],
     "Resource":[
          "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-snap
shot20130829"
         ]
   }
  ]
}
```

- IAM users in an AWS account with which a snapshot has been shared cannot perform actions on that snapshot unless they have IAM policies allowing those actions:
  - To list or describe a snapshot, they must have an IAM policy that allows the `DescribeClusterSnapshots` action. The following code shows an example:

```
{
  "Version": "2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "redshift:DescribeClusterSnapshots"
          ],
      "Resource":[
           "*"
          ]
    }
  ]
}
```

  - To restore a snapshot, users must have an IAM policy that allows the `RestoreFromClusterSnapshot` action and has a resource element that covers both the cluster they are attempting to create and the snapshot. For example, if a user in account `012345678912` has shared snapshot `my-snapshot20130829` with account `219876543210`, in order to create a cluster by restoring the snapshot, a user in account `219876543210` must have a policy such as the following:

```
{
  "Version": "2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "redshift:RestoreFromClusterSnapshot"
          ],
      "Resource":[
           "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-snap
shot20130829",
           "arn:aws:redshift:us-east-1:219876543210:cluster:from-another-
account"
          ]
    }
```

```
  ]
}
```

- Once access to a snapshot has been revoked from an AWS account, no users in that account can access the snapshot, even if they have IAM policies that allow actions on the previously shared snapshot resource.

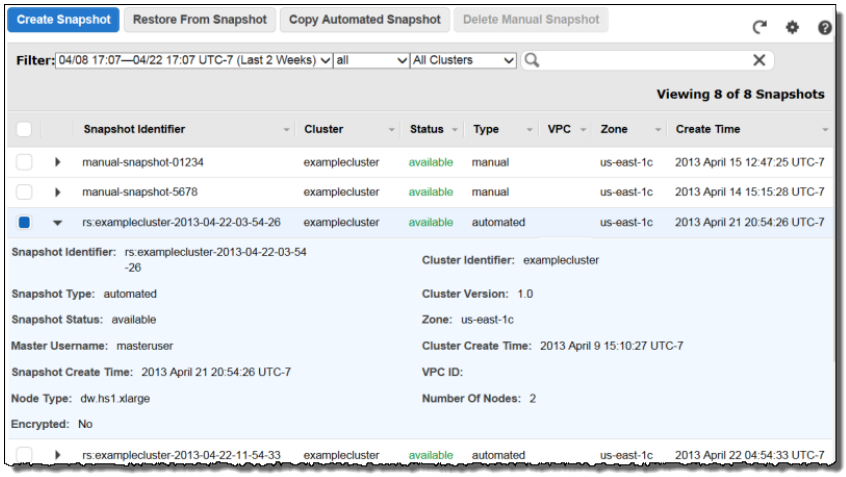
Managing Snapshots Using the Console

Topics

- Creating a Manual Snapshot (p. 87)
- Deleting a Manual Snapshot (p. 88)
- Copying an Automated Snapshot (p. 88)
- Restoring a Cluster from a Snapshot (p. 89)
- Sharing a Cluster Snapshot (p. 91)
- Configuring Cross-Region Snapshot Copy (p. 92)
- Modifying the Retention Period for Cross-Region Snapshot Copy (p. 93)
- Disabling Cross-Region Snapshot Copy (p. 94)

Amazon Redshift takes automatic, incremental snapshots of your data periodically and saves them to Amazon S3. Additionally, you can take manual snapshots of your data whenever you want. This section explains how to manage your snapshots from the Amazon Redshift console. For more information about snapshots, see Amazon Redshift Snapshots (p. 82).

All snapshot tasks in the Amazon Redshift console start from the snapshot list. You can filter the list by using the snapshot type, a time range, and the cluster associated with the snapshot. When you select an existing snapshot, the snapshot details are shown inline in the list, as shown in the example following. Depending on the snapshot type that you select, you will have different options available for working with the snapshot.

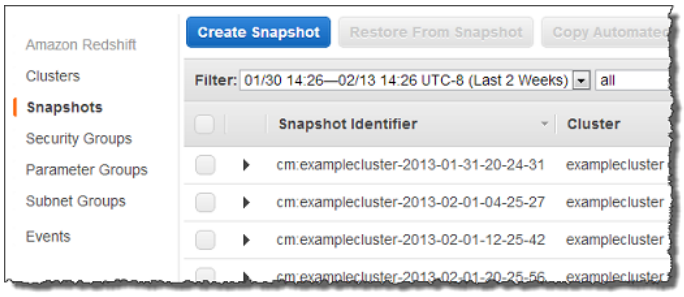

Create Snapshot
Restore From Snapshot
Copy Automated Snapshot
Delete Manual Snapshot
Filter: 04/08 17:07—04/22 17:07 UTC-7 (Last 2 Weeks) all All Clusters
Viewing 8 of 8 Snapshots
Snapshot Identifier Cluster Status Type VPC Zone Create Time
manual-snapshot-01234 examplecluster available manual us-east-1c 2013 April 15 12:47:25 UTC-7
manual-snapshot-5678 examplecluster available manual us-east-1c 2013 April 14 15:15:28 UTC-7
rs:examplecluster-2013-04-22-03-54-26 examplecluster available automated us-east-1c 2013 April 21 20:54:26 UTC-7
Snapshot Identifier: rs:examplecluster-2013-04-22-03-54-26
Cluster Identifier: examplecluster
Snapshot Type: automated
Cluster Version: 1.0
Snapshot Status: available
Zone: us-east-1c
Master Username: masteruser
Cluster Create Time: 2013 April 9 15:10:27 UTC-7
Snapshot Create Time: 2013 April 21 20:54:26 UTC-7
VPC ID:
Node Type: dw.hs1.xlarge
Number Of Nodes: 2
Encrypted: No
rs:examplecluster-2013-04-22-11-54-33 examplecluster available automated us-east-1c 2013 April 22 04:54:33 UTC-7

Creating a Manual Snapshot

You can create a manual snapshot of a cluster from the snapshots list as follows. Or, you can take a snapshot of a cluster in the cluster configuration pane. For more information, see Taking a Snapshot of a Cluster (p. 27).

To create a manual snapshot

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Snapshots**.
3. Click **Create Snapshot**.

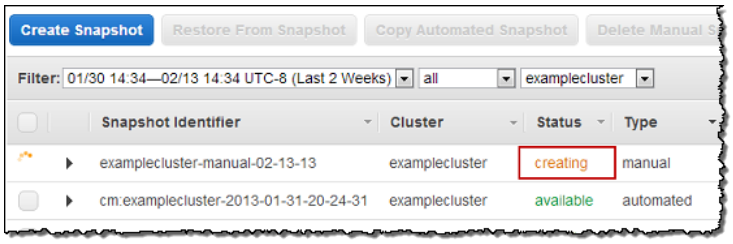


4. In the **Create Snapshot** dialog box, do the following:

   a. In the **Cluster Identifier** box, click the cluster that you want to take a snapshot of.
   b. In the **Snapshot Identifier** box, type a name for the snapshot.

5. Click **Create**.

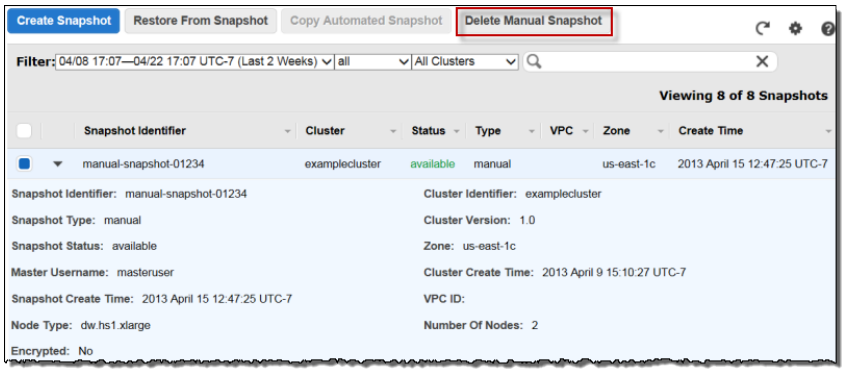
   The snapshot might take some time to complete. The new snapshot is displayed in the list of snapshots with its current status. The example following shows that `examplecluster-manual-02-13-13` is in the process of being created.



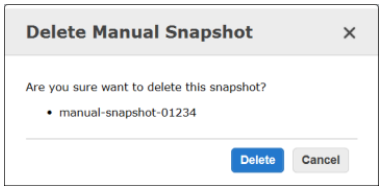
Deleting a Manual Snapshot

To delete a manual snapshot

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to delete, do any or all of the following:
   - In the **Time Range** box, click a time range that will narrow your search appropriately.
   - In the **Type** box, click **manual**.
   - In the **Cluster** box, click the cluster whose snapshot you want to delete.
4. In the snapshot list, click the row that contains the snapshot that you want to delete.
5. Click **Delete Manual Snapshot**.



6. In the **Delete Manual Snapshot** dialog box, click **Delete**.



Copying an Automated Snapshot

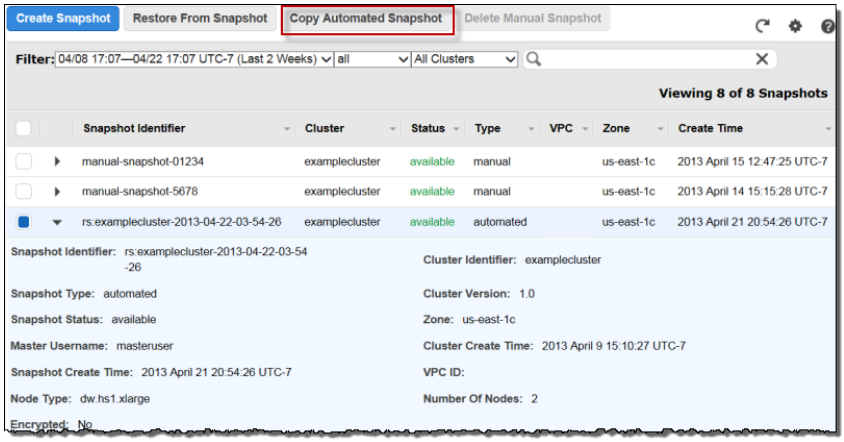
Automated snapshots are automatically deleted when their retention period expires, when you disable automated snapshots, or when you delete a cluster. If you want to keep an automated snapshot, you can copy it to a manual snapshot. Because Amazon Redshift never automatically deletes manual snapshots, you can keep this copy as long as you want.

To copy an automated snapshot

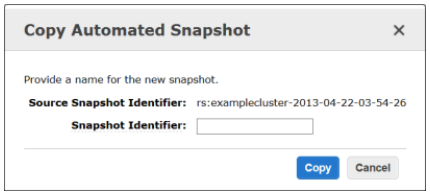
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to delete, do any or all of the following:

- In the **Time Range** box, click a time range that will narrow your search appropriately.
- In the **Type** box, click **automated**.
- In the **Cluster** box, click the cluster whose snapshot you want to copy.

4. In the snapshot list, click the row of the snapshot that you want to copy.
5. Click **Copy Automated Snapshot**.


Create Snapshot
Restore From Snapshot
Copy Automated Snapshot
Delete Manual Snapshot
Filter: 04/08 17:07—04/22 17:07 UTC-7 (Last 2 Weeks) all All Clusters
Viewing 8 of 8 Snapshots
Snapshot Identifier Cluster Status Type VPC Zone Create Time
manual-snapshot-01234 examplecluster available manual us-east-1c 2013 April 15 12:47:25 UTC-7
manual-snapshot-5678 examplecluster available manual us-east-1c 2013 April 14 15:15:28 UTC-7
rs:examplecluster-2013-04-22-03-54-26 examplecluster available automated us-east-1c 2013 April 21 20:54:26 UTC-7
Snapshot Identifier: rs:examplecluster-2013-04-22-03-54-26
Cluster Identifier: examplecluster
Snapshot Type: automated
Cluster Version: 1.0
Snapshot Status: available
Zone: us-east-1c
Master Username: masteruser
Cluster Create Time: 2013 April 9 15:10:27 UTC-7
Snapshot Create Time: 2013 April 21 20:54:26 UTC-7
VPC ID:
Node Type: dw.hs1.xlarge
Number Of Nodes: 2
Encrypted: No

6. In the **Snapshot Identifier** box of the **Copy Automated Snapshot** dialog box, type a name for the snapshot copy.



7. Click **Copy**.

Restoring a Cluster from a Snapshot

When you restore a cluster from a snapshot, Amazon Redshift creates a new cluster with all the snapshot data on the new cluster.

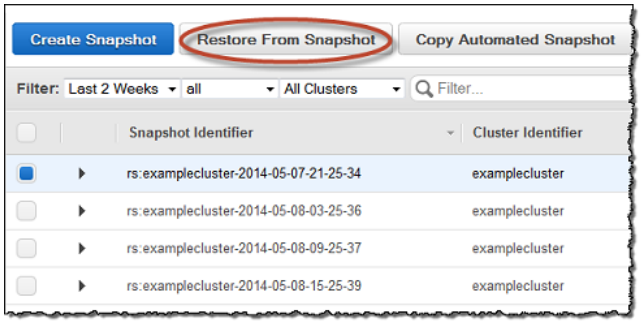
To restore a cluster from a snapshot

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to use, do any or all of the following:

   - In the **Time Range** box, click a time range that will narrow your search appropriately.
   - In the **Type** box, click **manual** or **automated**.
   - In the **Cluster** box, click the cluster whose snapshot you want to copy.

4. In the snapshot list, click the row that contains the snapshot that you want to use.

5. Click **Restore From Snapshot**.



6. In the **Restore Cluster from Snapshot** dialog box, do the following:

   a. In the **Cluster Identifier** box, type a cluster identifier for the restored cluster.

      Cluster identifiers must meet the following conditions:

      - They must contain from 1 to 255 alphanumeric characters or hyphens.
      - Alphabetic characters must be lowercase.
      - The first character must be a letter.
      - They cannot end with a hyphen or contain two consecutive hyphens.
      - They must be unique for all clusters within an AWS account.

   b. In the **Port** box, accept the port from the snapshot or change the value as appropriate.
   c. Select **Allow Version Upgrade** as appropriate.
   d. In **Cluster Subnet Group**, select the subnet group into which you want to restore the cluster.

      This option only appears if you restore the cluster into the EC2-VPC platform.
   e. In **Publicly Accessible**, select **Yes** if you want the cluster to have a public IP address that can be accessed over a public connection to the Internet, and select **No** if you want the cluster to have a private IP address that can only be accessed from within the VPC. If your AWS account allows you to create EC2-Classic clusters, the default is **No**. Otherwise, the default is **Yes**.

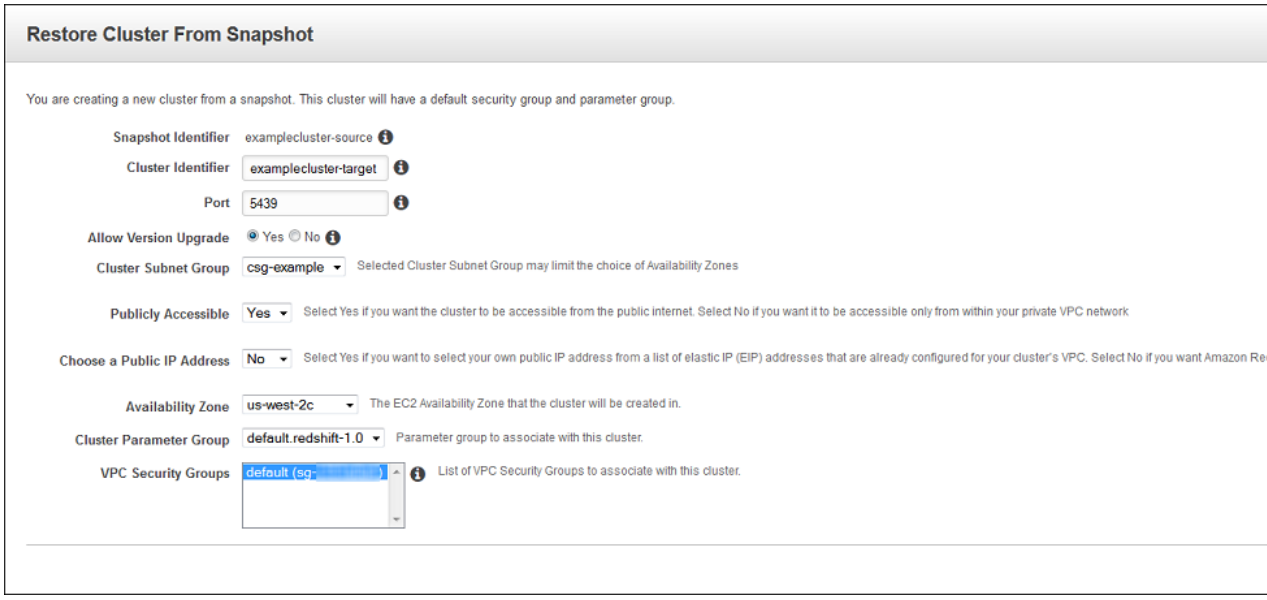
      This option only appears if you restore the cluster into the EC2-VPC platform.
   f. In **Choose a Public IP Address**, select **Yes** if you want to select an elastic IP (EIP) address that you already have configured. Otherwise, select **No** to have Amazon Redshift create an EIP for your instance.

      This option only appears if you restore the cluster into the EC2-VPC platform.
   g. In **Elastic IP**, select an EIP to use to connect to the cluster from outside of the VPC.

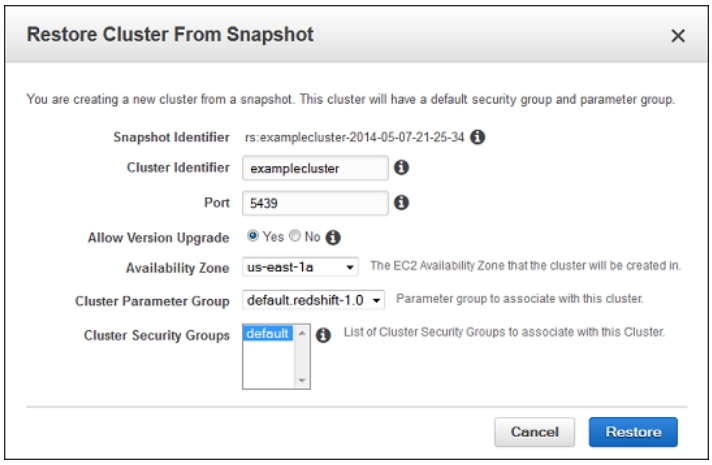
      This option only appears if you restore the cluster into the EC2-VPC platform and you select **Yes** in **Choose a Public IP Address**.
   h. In the **Availability Zone** box, accept the Availability Zone from the snapshot or change the value as appropriate.
   i. In **Cluster Parameter Group**, select a parameter group to associate with the cluster.
   j. In **Cluster Security Groups** or **VPC Security Groups**, select a security group to associate with the cluster. The types of security group that appear here depend on whether you're restoring the cluster into the EC2-Classic or EC2-VPC platform.

      The option to select a cluster security group or a VPC security group depends on whether you restore the cluster into the EC2-Classic platform or the EC2-VPC platform.

The following is an example of restoring a snapshot into a cluster that uses the EC2-VPC platform.


Restore Cluster From Snapshot
You are creating a new cluster from a snapshot. This cluster will have a default security group and parameter group.
Snapshot Identifier
examplecluster-source
Cluster Identifier
examplecluster-target
Port
5439
Allow Version Upgrade
Yes
No
Cluster Subnet Group
csg-example
Selected Cluster Subnet Group may limit the choice of Availability Zones
Publicly Accessible
Yes
Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network
Choose a Public IP Address
No
Availability Zone
us-west-2c
The EC2 Availability Zone that the cluster will be created in.
Cluster Parameter Group
default.redshift-1.0
Parameter group to associate with this cluster.
VPC Security Groups
List of VPC Security Groups to associate with this cluster.

The following is an example of restoring a snapshot into a cluster that uses the EC2-Classic platform.



7. Click **Restore**.

Sharing a Cluster Snapshot

You can authorize other users to access a manual snapshot you own, and you can later revoke that access when it is no longer required.

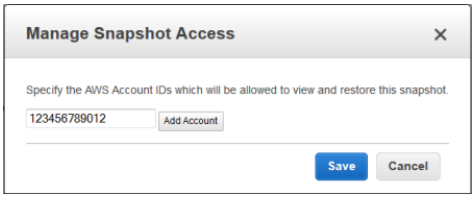
To share a cluster snapshot

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Snapshots**.
3. If you need to filter the list in order to find the snapshot that you want to delete, do any or all of the following:

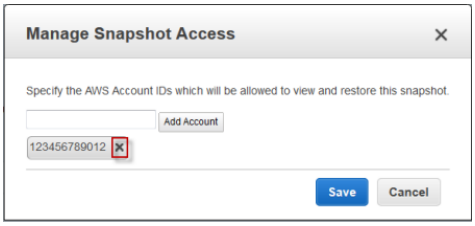
- In the **Time Range** box, click a time range that will narrow your search appropriately.
- In the **Type** box, click **manual** or **automated**.
- In the **Cluster** box, click the cluster whose snapshot you want to copy.

4. In the snapshot list, click the row that contains the snapshot that you want to use.
5. Click **Manage Access**.
6. In the **Manage Snapshot Access** dialog box, you can either authorize a user to access the snapshot or revoke a previously authorized access.

   - To authorize a user to access the snapshot, type that user's 12-digit AWS account ID in the box (omit the dashes), and then click **Add Account**.



   - To revoke the authorization for a user, click **X** beside that user's AWS account ID.



7. Click **Save** to save your changes, or **Cancel** to roll back the changes.

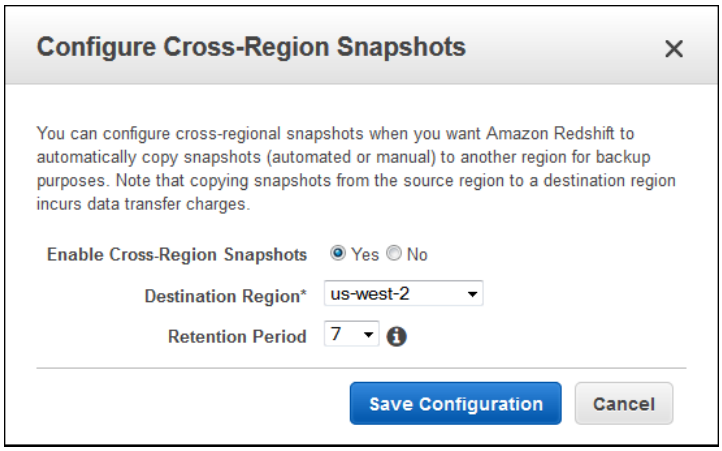
Configuring Cross-Region Snapshot Copy

You can configure Amazon Redshift to copy snapshots for a cluster to another region. To configure cross-region snapshot copy, you need to enable this copy feature and configure where to copy snapshots and how long to keep copied automated snapshots in the destination region.

To configure cross-region snapshot copy for a cluster

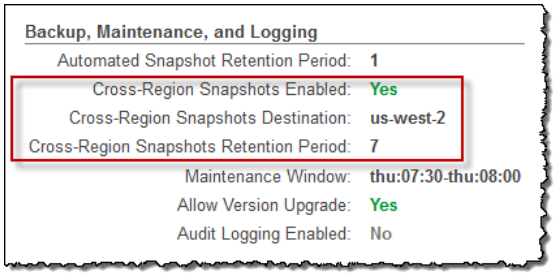
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**.
3. Click **Backup**, and then click **Configure Cross Region Snapshots**.
4. In the **Configure Cross Region Snapshots** dialog box, do the following:

   a. In the **Enable Cross Region Snapshots** box, click **Yes**.
   b. In the **Destination Region** box, select the region to which you want snapshots to be copied.
   c. In the **Retention Period** box, select the number of days that you want automated snapshots to be retained. After an automated snapshot reaches the limit in the retention period, it is deleted from the destination region.


Configure Cross-Region Snapshots
You can configure cross-regional snapshots when you want Amazon Redshift to automatically copy snapshots (automated or manual) to another region for backup purposes. Note that copying snapshots from the source region to a destination region incurs data transfer charges.
Enable Cross-Region Snapshots
Yes
No
Destination Region*
us-west-2
Retention Period
7
Save Configuration
Cancel

5. Click **Save Configuration**.

After you configure cross-region snapshot copy, the **Cluster** detail page updates with the information in the **Backup, Maintenance, and Logging** section.


Backup, Maintenance, and Logging
Automated Snapshot Retention Period: 1
Cross-Region Snapshots Enabled: Yes
Cross-Region Snapshots Destination: us-west-2
Cross-Region Snapshots Retention Period: 7
Maintenance Window: thu:07:30-thu:08:00
Allow Version Upgrade: Yes
Audit Logging Enabled: No

Modifying the Retention Period for Cross-Region Snapshot Copy

After you configure cross-region snapshot copy, you might want to change the settings. You can easily change the retention period by selecting a new number of days and saving the changes.

> **Caution**
> You cannot modify the destination region after cross-region snapshot copy is configured. If you want to copy snapshots to a different region, you must first disable cross-region snapshot copy, and then re-enable it with a new destination region and retention period. Because any copied automated snapshots are deleted after you disable cross-region snapshot copy, you should determine if there are any that you want to keep and copy them to manual snapshots before disabling cross-region snapshot copy.

To modify the retention period for snapshots copied to a destination cluster

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**.
3. Click **Backup**, and then click **Configure Cross Region Snapshots**.
4. In the **Retention Period** box, select the new number of days that you want automated snapshots to be retained in the destination region.

If you select a smaller number of days to retain snapshots in the destination region, any automated snapshots that were taken before the new retention period will be deleted. If you select a larger number of days to retain snapshots in the destination region, the retention period for existing automated snapshots will be extended by the difference between the old value and the new value.

5. Click **Save Configuration**.

Disabling Cross-Region Snapshot Copy

You can disable cross-region snapshot copy for a cluster when you no longer want Amazon Redshift to copy snapshots to a destination region.

To disable cross-region snapshot copy for a cluster

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**.
3. Click **Backup**, and then click **Configure Cross Region Snapshots** to open the **Configure Cross Region Snapshots** dialog box.
4. In the **Enable Cross Region Snapshots** box, click **No**.
5. Click **Save Configuration**.

Managing Snapshots Using the AWS SDK for Java

The following example demonstrates these common operations involving a snapshot:

- Creating a manual cluster snapshot of a cluster.
- Displaying information about all the snapshots of a cluster.
- Deleting manual snapshots of a cluster.

In this example, a snapshot of the cluster is initiated. When the snapshot is successfully created, all manual snapshots for the cluster that were created before the new snapshot are deleted. When creation of the manual snapshot is initiated, the snapshot is not immediately available. Therefore, this example uses a loop to poll for the status of the snapshot by calling the `describeClusterSnapshot` method. It normally takes a few moments for a snapshot to become available after initiation. For more information about snapshots, see Amazon Redshift Snapshots (p. 82).

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and provide a cluster identifier.

```
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.CreateClusterSnapshotRequest;
import com.amazonaws.services.redshift.model.DeleteClusterSnapshotRequest;
import com.amazonaws.services.redshift.model.DescribeClusterSnapshotsRequest;
```

```
import com.amazonaws.services.redshift.model.DescribeClusterSnapshotsResult;
import com.amazonaws.services.redshift.model.Snapshot;

public class CreateAndDescribeSnapshot {

    public static AmazonRedshiftClient client;
   public static String clusterIdentifier = "***provide cluster identifier***";

    public static long sleepTime = 10;

    public static void main(String[] args) throws IOException {

        AWSCredentials credentials = new PropertiesCredentials(
                CreateAndDescribeSnapshot.class
                        .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             // Unique snapshot identifier
             String snapshotId = "my-snapshot-" + (new SimpleDateFormat("yyyy-
MM-dd-HH-mm-ss")).format(new Date());

             Date createDate = createManualSnapshot(snapshotId);
             waitForSnapshotAvailable(snapshotId);
             describeSnapshots();
             deleteManualSnapshotsBefore(createDate);
             describeSnapshots();

        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static Date createManualSnapshot(String snapshotId) {

       CreateClusterSnapshotRequest request = new CreateClusterSnapshotRequest()

        .withClusterIdentifier(clusterIdentifier)
        .withSnapshotIdentifier(snapshotId);
        Snapshot snapshot = client.createClusterSnapshot(request);
        System.out.format("Created cluster snapshot: %s\n", snapshotId);
        return snapshot.getSnapshotCreateTime();
    }

    private static void describeSnapshots() {

       DescribeClusterSnapshotsRequest request = new DescribeClusterSnapshots
Request()
            .withClusterIdentifier(clusterIdentifier);
        DescribeClusterSnapshotsResult result = client.describeClusterSnap
shots(request);

        printResultSnapshots(result);
    }

    private static void deleteManualSnapshotsBefore(Date creationDate) {
```

```
       DescribeClusterSnapshotsRequest request = new DescribeClusterSnapshots
Request()
            .withEndTime(creationDate)
            .withClusterIdentifier(clusterIdentifier)
            .withSnapshotType("manual");

        DescribeClusterSnapshotsResult result = client.describeClusterSnap
shots(request);

        for (Snapshot s : result.getSnapshots()) {
           DeleteClusterSnapshotRequest deleteRequest = new DeleteClusterSnap
shotRequest()
            .withSnapshotIdentifier(s.getSnapshotIdentifier());
          Snapshot deleteResult = client.deleteClusterSnapshot(deleteRequest);

            System.out.format("Deleted snapshot %s\n", deleteResult.getSnap
shotIdentifier());
        }
    }

    private static void printResultSnapshots(DescribeClusterSnapshotsResult
result) {
        System.out.println("\nSnapshot listing:");
        for (Snapshot snapshot : result.getSnapshots()) {
           System.out.format("Identifier: %s\n", snapshot.getSnapshotIdentifi
er());
          System.out.format("Snapshot type: %s\n", snapshot.getSnapshotType());

            System.out.format("Snapshot create time: %s\n", snapshot.getSnap
shotCreateTime());
           System.out.format("Snapshot status: %s\n\n", snapshot.getStatus());

        }
    }

    private static Boolean waitForSnapshotAvailable(String snapshotId) throws
InterruptedException {
        Boolean snapshotAvailable = false;
        System.out.println("Wating for snapshot to become available.");
        while (!snapshotAvailable) {
           DescribeClusterSnapshotsResult result = client.describeClusterSnap
shots(new DescribeClusterSnapshotsRequest()
                .withSnapshotIdentifier(snapshotId));
            String status = (result.getSnapshots()).get(0).getStatus();
            if (status.equalsIgnoreCase("available")) {
                snapshotAvailable = true;
            }
            else {
                System.out.print(".");
                Thread.sleep(sleepTime*1000);
            }
        }
        return snapshotAvailable;
    }

}
```

Managing Snapshots Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage snapshots.

- authorize-snapshot-access
- copy-cluster-snapshot
- create-cluster-snapshot
- delete-cluster-snapshot
- describe-cluster-snapshots
- disable-snapshot-copy
- enable-snapshot-copy
- modify-snapshot-copy-retention-period
- restore-from-cluster-snapshot
- revoke-snapshot-access

You can use the following Amazon Redshift API actions to manage snapshots.

- AuthorizeSnapshotAccess
- CopyClusterSnapshot
- CreateClusterSnapshot
- DeleteClusterSnapshot
- DescribeClusterSnapshots
- DisableSnapshotCopy
- EnableSnapshotCopy
- ModifySnapshotCopyRetentionPeriod
- RestoreFromClusterSnapshot
- RevokeSnapshotAccess

For more information about Amazon Redshift snapshots, see Amazon Redshift Snapshots (p. 82).

Amazon Redshift Database Encryption

Topics

- Overview (p. 98)
- Hardware Security Modules (p. 99)
- Rotating Encryption Keys (p. 105)

Overview

Amazon Redshift provides database encryption for its clusters to help protect data at rest. When you enable encryption for your cluster, Amazon Redshift encrypts all data by using hardware-accelerated AES-256. This encryption includes data blocks and system metadata, and it applies to both the active cluster and any cluster backups.

Amazon Redshift uses a four-tier, key-based architecture for encryption. These keys consist of data encryption keys, a database key, a cluster key, and a master key.

Data encryption keys encrypt data blocks in the cluster. Each data block is assigned a randomly-generated AES-256 key. These keys are encrypted by using the database key for the cluster.

The *database key* encrypts data encryption keys in the cluster. The database key is a randomly-generated AES-256 key. It is stored on disk in a separate network from the Amazon Redshift cluster and encrypted by the cluster key. Amazon Redshift passes the database key across a secure channel and keeps it in memory in the cluster.

The *cluster key* encrypts the database key for the Amazon Redshift cluster. You can use either AWS Key Management Service (AWS KMS) or a hardware security module (HSM) to manage the cluster key.

The *master key* encrypts the cluster key if it is managed by AWS KMS. The master key encrypts the cluster-key-encrypted database key if the cluster key is stored in an HSM. For more information about AWS KMS, go to AWS Key Management Service Developer Guide. For more information about HSM, see Hardware Security Modules (p. 99).

Though encryption is optional, we recommend using it to help protect sensitive data at rest, and it might be required depending on the business, privacy, or security rules that apply to the data that you store. For example, the Payment Card Industry Data Security Standard (PCI DSS), the Sarbanes-Oxley Act

(SOX), the Health Insurance Portability and Accountability Act (HIPAA), and other such regulations provide guidelines for handling specific types of data.

Encryption is an immutable property of the cluster. The only way to go from an encrypted to a nonencrypted cluster or vice versa is to unload the data and reload it to a new cluster. Encryption also applies to backups. When restoring from an encrypted snapshot, the new cluster will be encrypted as well.

Note
When you enable encryption in your cluster, it will have an impact on performance, even though it is hardware-accelerated. On average, we expect you will see approximately a 20 percent degradation, with peak overheads of 40 percent. You should take this into account when deciding whether you should enable encryption when you create the cluster.

Hardware Security Modules

You can use a hardware security module (HSM) to generate and manage your Amazon Redshift cluster key. HSMs are devices that provide direct control of key generation and management. They provide greater security by separating key management from the application and database layers. Amazon Redshift supports both AWS CloudHSM and on-premises HSMs for key management.

When you configure your cluster to use an HSM, Amazon Redshift sends a request to the HSM to create a cluster key. The HSM uses the cluster key to encrypt the database key. The cluster key is stored in the HSM. The cluster key decrypts the encrypted database key, and then the unencrypted database key is passed over a secure channel to the cluster, where it is loaded into memory. The database key is then used to encrypt all of the data encryption keys that encrypt data blocks.

When you opt to use an HSM for management of your cluster key, you need to configure a trusted network link between Amazon Redshift and your HSM. Doing this requires configuration of client and server certificates.

Amazon Redshift creates a public client certificate from a randomly generated private and public key pair. These are encrypted and stored internally. You download and register the public client certificate in your HSM, and assign it to the applicable HSM partition.

You provide Amazon Redshift with the HSM IP address, HSM partition name, HSM partition password, and a public HSM server certificate, which is encrypted by using an internal master key. Amazon Redshift completes the configuration process and verifies that it can connect to the HSM. If it cannot, the cluster is put into INCOMPATIBLE_HSM state and the cluster is not created. In this case, you must delete the incomplete cluster and try again.

After initial configuration, if Amazon Redshift fails to connect to the HSM, an event is logged. For more information about these events, see Amazon Redshift Event Notifications (p. 202)

Configuring HSM Using the Amazon Redshift Console

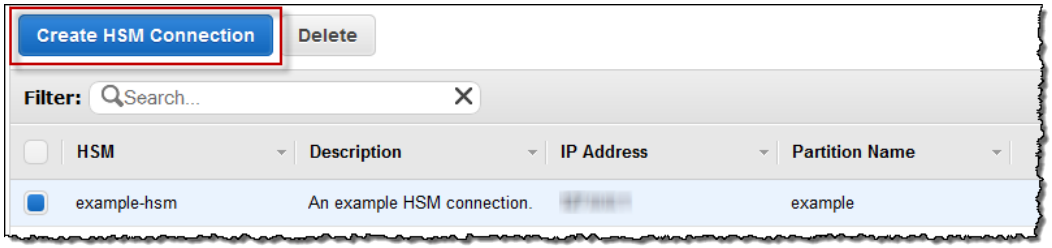
Topics

- Creating an HSM Connection (p. 100)
- Creating an HSM Client Certificate (p. 101)
- Displaying the Public Key for an HSM Client Certificate (p. 104)
- Deleting an HSM Connection (p. 104)
- Deleting an HSM Client Certificate (p. 104)

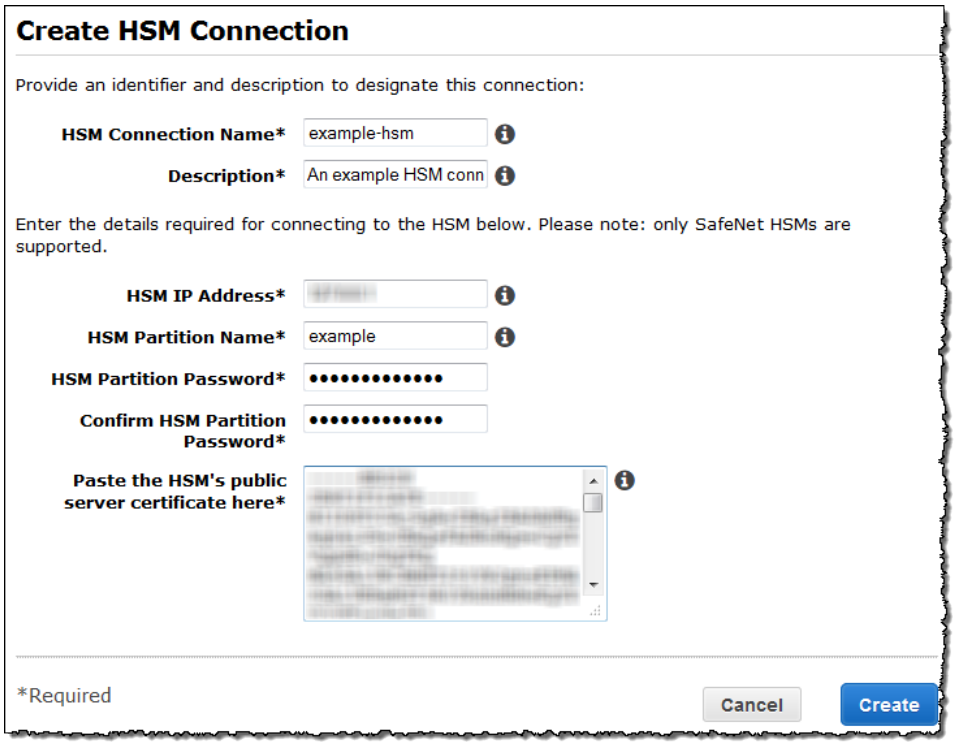
You can use the following procedures to configure hardware security modules by using the AWS Management Console.

Creating an HSM Connection

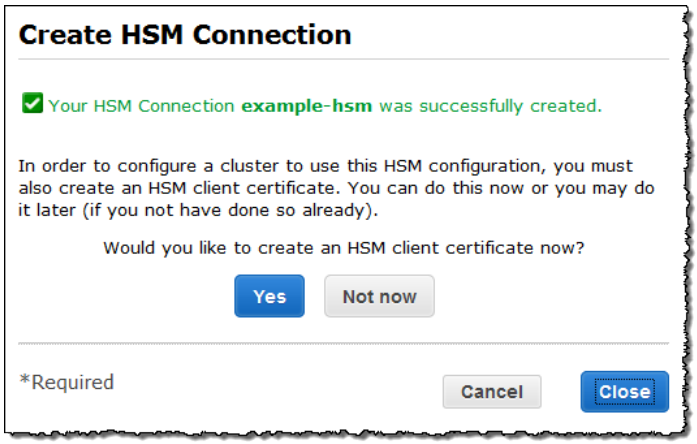
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Security**, and then click the **HSM Connections** tab.
3. Click **Create HSM Connection.**



4. On the **Create HSM Connection** page, type the following information:
   a. In the **HSM Connection Name** box, type a name to identify this connection.
   b. In the **Description** box, type a description about the connection.
   c. In the **HSM IP Address** box, type the IP address for your HSM.
   d. In the **HSM Partition Name** box, type the name of the partition that Amazon Redshift should connect to.
   e. In the **HSM Partition Password** box, type the password that is required to connect to the HSM partition.
   f. Copy the public server certificate from your HSM and paste it in the **Paste the HSM's public server certificate here** box.
   g. Click **Create**.

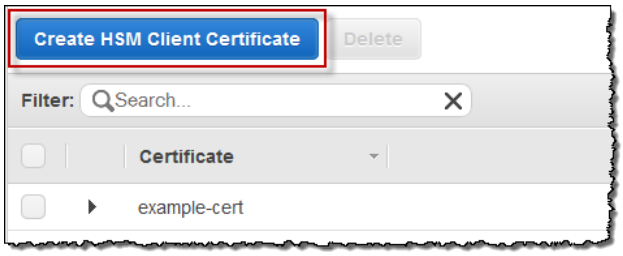


5. After the connection is created, you can create an HSM client certificate. If you want to create an HSM client certificate immediately after creating the connection, click **Yes** and complete the steps in the next procedure. Otherwise, click **Not now** to return to the list of HSM connections and complete the remainder of the process at another time.


Create HSM Connection
Your HSM Connection example-hsm was successfully created.
In order to configure a cluster to use this HSM configuration, you must also create an HSM client certificate. You can do this now or you may do it later (if you not have done so already).
Would you like to create an HSM client certificate now?
Yes
Not now
*Required
Cancel
Close

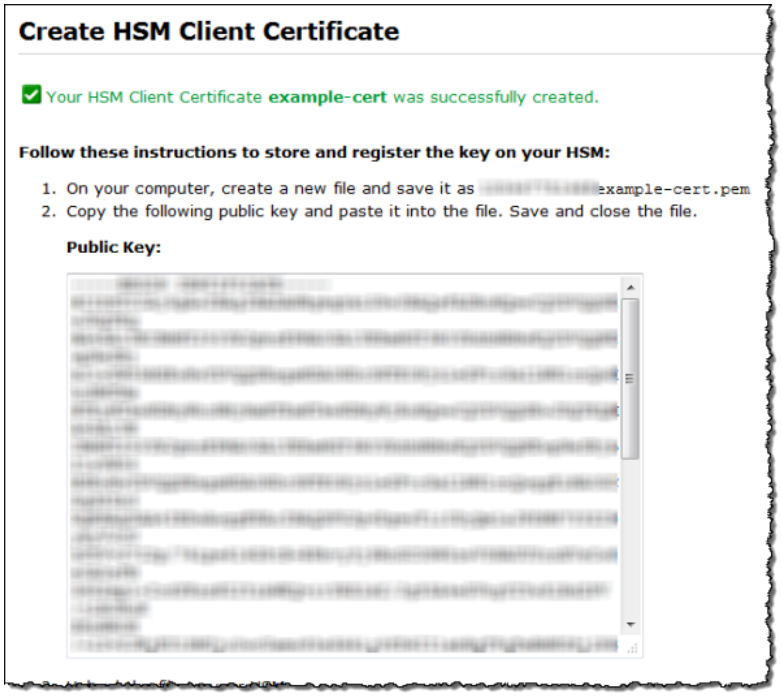
Creating an HSM Client Certificate

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Security**, and then click the **HSM Certificates** tab.
3. Click **Create HSM Client Certificate**.



4. On the **Create HSM Client Certificate** page, type a name in the **HSM Client Certificate Identifier** box to identify this client certificate.

5. Click **Next**.
6. After the certificate is created, a confirmation page appears with information to register the key on your HSM. If you do not have permission to configure the HSM, coordinate the following steps with an HSM administrator.


Create HSM Client Certificate
Your HSM Client Certificate example-cert was successfully created.
Follow these instructions to store and register the key on your HSM:
1. On your computer, create a new file and save it as example-cert.pem
2. Copy the following public key and paste it into the file. Save and close the file.
Public Key:

   a. On your computer, open a new text file.
   b. In the Amazon Redshift console, on the **Create HSM Client Certificate** confirmation page, copy the public key.

c. Paste the public key into the open file and save it with the file name displayed in step 1 from the confirmation page. Make sure that you save the file with the `.pem` file extension, for example: `123456789mykey.pem`.

d. Upload the `.pem` file to your HSM.

e. On the HSM, open a command-prompt window and run the commands listed in step 4 on the confirmation page to register the key. The command uses the following format, with *`ClientName`*, *`KeyFilename`*, and *`PartitionName`* being values you need to replace with your own:

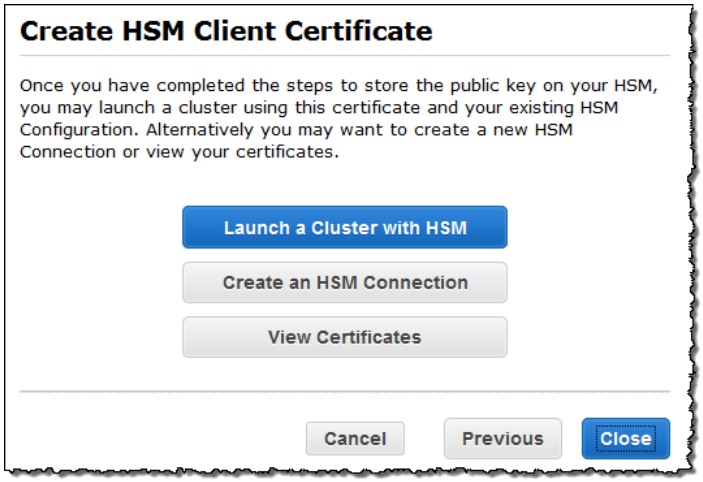
```
client register -client ClientName -hostname KeyFilename

client assignPartition -client ClientName -partition PartitionName
```

For example:

```
client register -client MyClient -hostname 123456789mykey

client assignPartition -client MyClient -partition MyPartition
```

f. After you register the key on the HSM, click **Next**.

7. After the HSM client certificate is created and registered, click one of the following buttons:


Create HSM Client Certificate
Once you have completed the steps to store the public key on your HSM, you may launch a cluster using this certificate and your existing HSM Configuration. Alternatively you may want to create a new HSM Connection or view your certificates.
Launch a Cluster with HSM
Create an HSM Connection
View Certificates
Cancel
Previous
Close

- **Launch a Cluster with HSM**. This option starts the process of launching a new cluster. During the process, you can select an HSM to store encryption keys. For more information about the launch cluster process, see Managing Clusters Using the Console (p. 13).

  Create an HSM Connection. This option starts the **Create HSM Connection** process.

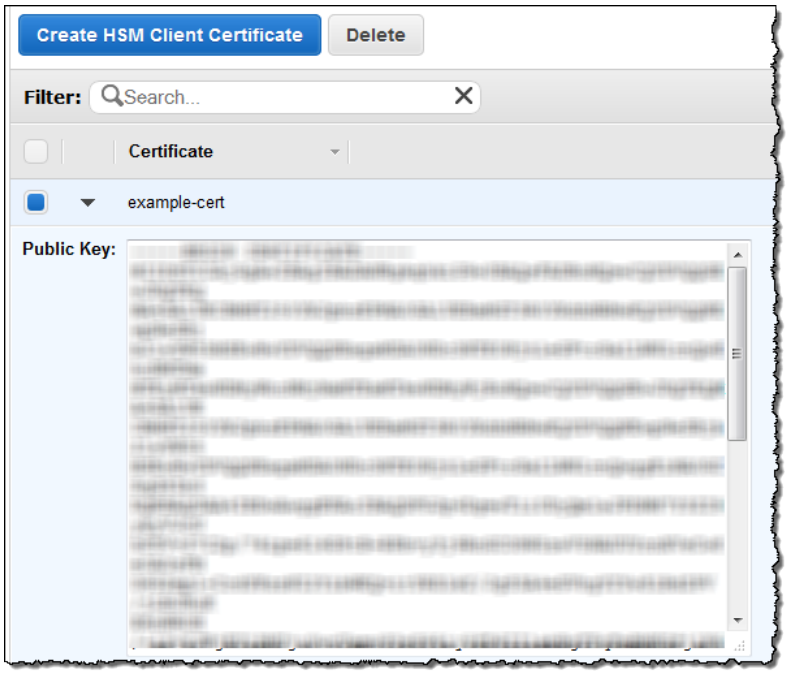
  View Certificates. This option returns you to **HSM** in the navigation pane and displays a list of client certificates on the **Certificates** tab.

  Previous. This option returns you to the **Create HSM Client Certificates** confirmation page.

  Close. This option returns you to **HSM** in the navigation pane and displays a list of HSM connections on the **Connections** tab.

Displaying the Public Key for an HSM Client Certificate

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security**, and then click the **HSM Certificates** tab.
3. Click the HSM client certificate to display the public key. This key is the same one that you added to the HSM in the procedure preceding procedure, Creating an HSM Client Certificate (p. 101)



Deleting an HSM Connection

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Security**, and then click the **HSM Connections** tab.
3. Click the HSM connection that you want to delete.
4. In the **Delete HSM Connection** dialog box, click **Delete** to delete the connection from Amazon Redshift, or click **Cancel** to return to the **HSM Connections** tab without deleting the connection.

Deleting an HSM Client Certificate

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Security** and select the **HSM Certificates** tab.
3. In the list, click the HSM client certificate that you want to delete.
4. In the **Delete HSM Client Certificate** dialog box, click **Delete** to delete the certificate from Amazon Redshift, or click **Cancel** to return to the **Certificates** tab without deleting the certificate.

Configuring HSM Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage hardware security modules.

- create-hsm-client-certificate
- create-hsm-configuration
- delete-hsm-client-certificate
- delete-hsm-configuration
- describe-hsm-client-certificates
- describe-hsm-configurations

You can use the following Amazon Redshift API actions to manage hardware security modules.

- CreateHsmClientCertificate
- CreateHsmConfiguration
- DeleteHsmClientCertificate
- DeleteHsmConfiguration
- DescribeHsmClientCertificates
- DescribeHsmConfigurations

Rotating Encryption Keys

In Amazon Redshift, you can rotate encryption keys for encrypted clusters. When you start the key rotation process, Amazon Redshift rotates the cluster encryption key and database encryption key for the specified cluster, and for any of the cluster's manual and automatic snapshots. The cluster is put into a ROTATING_KEYS state until the rotation completes, at which time it returns to the AVAILABLE state. Amazon Redshift handles decryption and re-encryption during the key rotation process.

> **Note**
> You cannot rotate keys for snapshots without a source cluster. Before you delete a cluster, consider whether its snapshots rely on key rotation.

Because the cluster is momentarily unavailable during the key rotation process, you should rotate keys only as often as your data needs require or when you suspect the keys might have been compromised. As a best practice, you should review the type of data that you store and plan how often to rotate the keys. The frequency for rotating keys varies depending on your corporate policies for data security, and any industry standards regarding sensitive data and regulatory compliance. Ensure that your plan balances security needs with availability considerations for your cluster.

Rotating Encryption Keys Using the Amazon Redshift Console

Topics

You can use the following procedure to rotate encryption keys by using the AWS Management Console.

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.

2. In the navigation pane, click **Clusters**.
3. In the list, click the cluster for which you want to rotate keys.
4. Click **Database**, and then click **Rotate Encryption Keys**.
5. Click **Yes, Rotate Keys** if you want to rotate the keys or **Cancel** if you do not.

   Note
   Your cluster will be momentarily unavailable until the key rotation process completes.

Rotating Encryption Keys Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to rotate encryption keys.

- rotate-encryption-key

You can use the following Amazon Redshift API operations to rotate encryption keys.

- RotateEncryptionKey

Purchasing Amazon Redshift Reserved Nodes

Topics

- Overview (p. 107)
- Purchasing a Reserved Node Offering with the Amazon Redshift Console (p. 108)
- Purchasing a Reserved Node Offering Using the AWS SDK for Java (p. 111)
- Purchasing a Reserved Node Offering Using the Amazon Redshift CLI and API (p. 114)

Overview

When you create an Amazon Redshift cluster, you specify information such the AWS region where you want Amazon Redshift to create the cluster, the number of nodes you want in the cluster, and the node type. Amazon Redshift provisions the requested compute nodes as soon as you request them, and your AWS account is billed the hourly rate for the provisioned compute nodes.

If you intend to keep your cluster running continuously for a prolonged period, say, one year or more, you can pay considerably less by reserving the compute nodes for a one-year or three-year period. To reserve compute nodes, you purchase what are called reserved node offerings. You purchase one offering for each compute node that you want to reserve.

When you reserve a compute node, you pay a fixed up-front charge and then an hourly recurring charge, whether your cluster is running or not. The hourly charges, however, are significantly lower than those for on-demand usage. For detailed pricing information, go to the Amazon Redshift product detail page.

What Are Reserved Node Offerings?

To reserve a compute node, you choose a node type and the duration for which you want to reserve the compute nodes. You then specify the utilization type, the anticipated utilization of the node, which influences pricing. At present, Amazon Redshift only offers "Heavy Utilization" reserved node offerings.

You can purchase a reserved node offering for an existing Amazon Redshift cluster. If you do so, remember that the node type and region of your reserved node must match those for your existing cluster. If these values don't match, Amazon Redshift cannot bill the cluster nodes at the lower reserved node rate, and so your cluster nodes will continue to accrue charges at the on-demand rate. In addition, your AWS

account will be billed for the up-front and recurring charges of the reserved nodes that you purchased. Amazon Redshift has a predefined list of the reserved node offerings that are available for purchase. Each offering describes the node type, the duration for which it will be reserved, offering type, and fixed and recurring charges. From the Amazon Redshift console, you select these values. You can also obtain a list of available offerings by using the Amazon Redshift API. Each offering has a unique ID that you can use to identify the one that you intend to purchase.

To purchase a reserved node, you pick a reserved node offering that matches your cluster's node type and region. You can also specify the quantity of reserved node offerings that you'd like to purchase. If you are purchasing a reserved node by programmatically, the region is determined by the endpoint that you are using. If you are purchasing a reserved node by using the Amazon Redshift console, you can use the region selector.

When you purchase reserved node offerings, Amazon Redshift does not allocate any new resources for your account. You must still create an Amazon Redshift cluster, but ensure that the cluster you create matches the node-type and region of the reserved nodes you've purchased.

Note that the purchasing a reserved node offering is only a billing concept. If you are currently running any on-demand nodes that match the description of the nodes you reserved, we will immediately start billing usage of those nodes at the reserved node rate. Also any similar nodes that you run in excess of the number of nodes you reserved will be billed at the normal on-demand rate. Let's look at the following examples:

- Suppose you have purchased three reserved node offerings of a specific type in a specific AWS region. Now suppose you create two clusters in the same region, one with three nodes and another with two nodes that use the same node type. After provisioning the clusters, Amazon Redshift will bill you the reserved node rate for the first three nodes and on-demand hourly rate for the other two nodes.
- Suppose you already have a cluster running with two nodes and you are paying the on-demand hourly rate. You then decided to use the cluster for a prolonged period, and purchased two reserved node offerings. Your existing cluster will continue to function, except that Amazon Redshift will now bill you the reserved node rate instead of hourly on-demand rate.

Because you purchase reserved nodes based on a node type and region, the existing reserved pricing rate no longer applies if you resize your cluster into a different node type. After resize, on-demand pricing applies to the nodes in the new cluster unless you have purchased reserved nodes that match the node type of the new cluster and there are nodes of that type available.

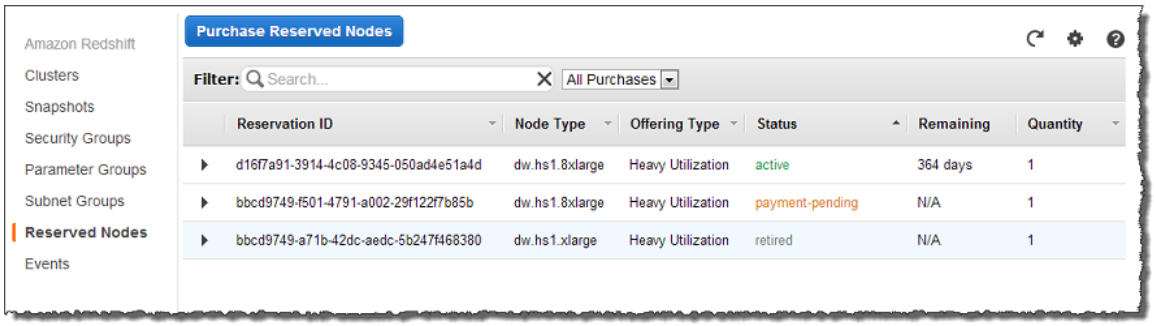
Purchasing a Reserved Node Offering with the Amazon Redshift Console

Topics

- Listing Reserved Node Offerings (p. 109)
- Purchasing Reserved Node Offerings (p. 109)

This section explains how to purchase a reserved node offering and to view your existing reserved node offerings by using the Amazon Redshift console. The reserved nodes list provides details about your offerings. It is also where you go to purchase new nodes. When you select an offering in the list, details of the offering are displayed.

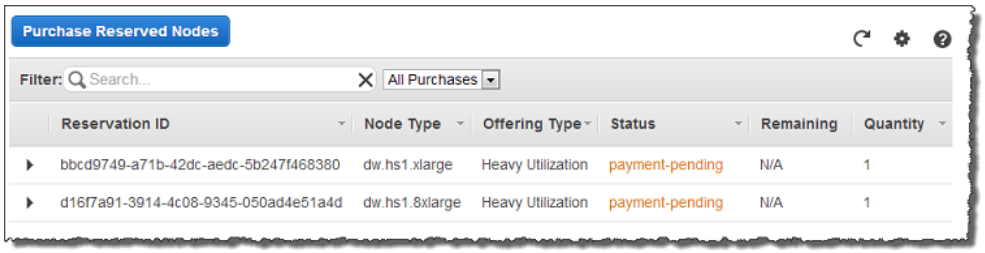
The status of an offering may be `payment-pending`, `active`, `payment-failed` or `retired`. An offer can not be used until the offering is active.



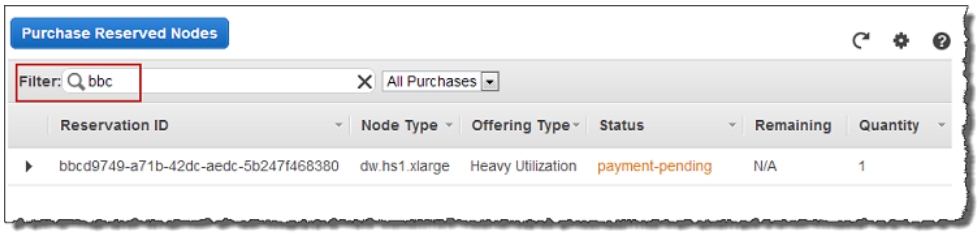
Listing Reserved Node Offerings

To list reserved node offerings

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Reserved Nodes**.



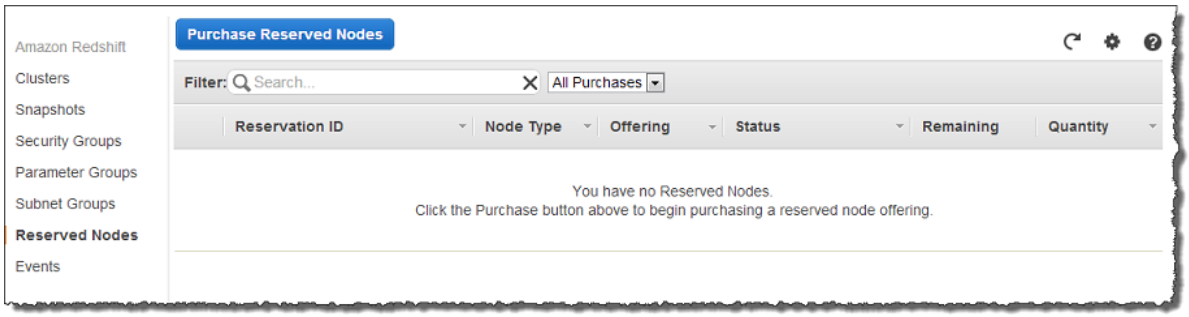
3. (Optional) To filter the display of offers, use the **Filter** boxes. Do one or both of the following:
   - Type a full or partial reservation ID. The displayed items will be only offers that include the string you type.
   - Click **All Purchases**, **Active**, or **Inactive**.



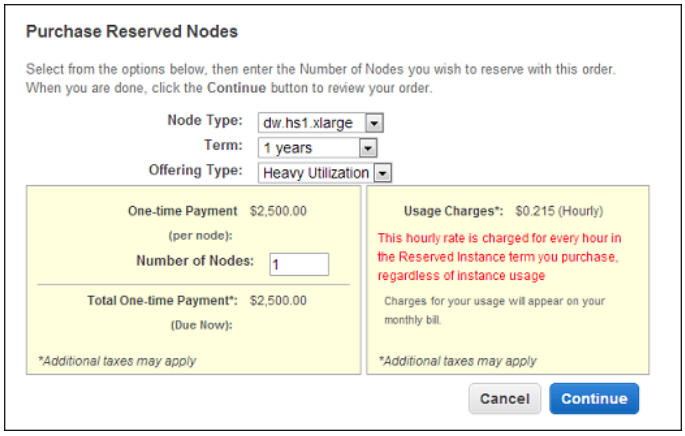
Purchasing Reserved Node Offerings

To purchase a reserved node offering

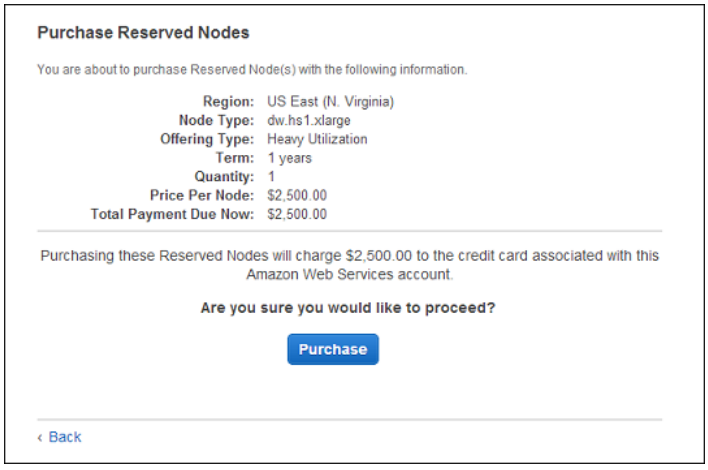
1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Reserved Nodes**.
3. Click **Purchase Reserved Nodes**.



4. In the **Purchase Reserved Nodes** wizard, specify **Node Type**, **Term**, **Offering Type**, and **Number of Nodes**.

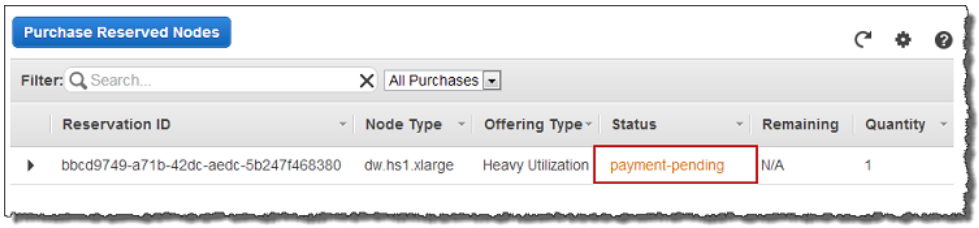

Purchase Reserved Nodes
Select from the options below, then enter the Number of Nodes you wish to reserve with this order.
When you are done, click the Continue button to review your order.
Node Type: dw.hs1.xlarge
Term: 1 years
Offering Type: Heavy Utilization
One-time Payment (per node): $2,500.00
Number of Nodes: 1
Total One-time Payment*: $2,500.00
(Due Now):
*Additional taxes may apply
Usage Charges*: $0.215 (Hourly)
This hourly rate is charged for every hour in the Reserved Instance term you purchase, regardless of instance usage
Charges for your usage will appear on your monthly bill.
*Additional taxes may apply
Cancel
Continue

5. When the settings are as you want them, click **Continue**.
6. Review the offering details, and then click **Purchase**.


Purchase Reserved Nodes
You are about to purchase Reserved Node(s) with the following information.
Region: US East (N. Virginia)
Node Type: dw.hs1.xlarge
Offering Type: Heavy Utilization
Term: 1 years
Quantity: 1
Price Per Node: $2,500.00
Total Payment Due Now: $2,500.00
Purchasing these Reserved Nodes will charge $2,500.00 to the credit card associated with this Amazon Web Services account.
Are you sure you would like to proceed?
Purchase
‹ Back

7. In the reserved nodes list, check the **Status** of your offering.

   When the status changes to `active`, you can use your node. At that time, the **Remaining Days** box will display the correct value.



Purchasing a Reserved Node Offering Using the AWS SDK for Java

The following example demonstrates how to use the AWS SDK for Java to do the following:

- List existing reserved nodes.
- Search for a new reserved node offering based on specified node criteria.
- Purchase a reserved node.

In this example, all the reserved node offerings that match a specified node type and fixed price value are selected. Then, the program goes through each found offering and allows you to purchase the offering.

> **Important**
> If you run this program and accept the offer to purchase a reserved node offering, you will be charged for the offering.

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and provide a different node type and fixed price.

```
import java.io.DataInput;
import java.io.DataInputStream;
import java.io.IOException;
import java.util.ArrayList;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.DescribeReservedNodeOfferings
Request;
import com.amazonaws.services.redshift.model.DescribeReservedNodeOfferingsResult;
import com.amazonaws.services.redshift.model.DescribeReservedNodesResult;
import com.amazonaws.services.redshift.model.PurchaseReservedNodeOfferingRequest;
import com.amazonaws.services.redshift.model.ReservedNode;
import com.amazonaws.services.redshift.model.ReservedNodeAlreadyExistsException;
import com.amazonaws.services.redshift.model.ReservedNodeOffering;
import com.amazonaws.services.redshift.model.ReservedNodeOfferingNotFoundExcep
tion;
import com.amazonaws.services.redshift.model.ReservedNodeQuotaExceededException;


public class ListAndPurchaseReservedNodeOffering {

    public static AmazonRedshiftClient client;
```

```
    public static String nodeTypeToPurchase = "dw.hs1.xlarge";
    public static Double fixedPriceLimit = 10000.00;
    public static ArrayList<ReservedNodeOffering> matchingNodes = new ArrayL
ist<ReservedNodeOffering>();

    public static void main(String[] args) throws IOException {


        AWSCredentials credentials = new PropertiesCredentials(
                ListAndPurchaseReservedNodeOffering.class
                .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             listReservedNodes();
             findReservedNodeOffer();
             purchaseReservedNodeOffer();

        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void listReservedNodes() {
        DescribeReservedNodesResult result = client.describeReservedNodes();
        System.out.println("Listing nodes already purchased.");
        for (ReservedNode node : result.getReservedNodes()) {
            printReservedNodeDetails(node);
        }
    }

    private static void findReservedNodeOffer()
    {
        DescribeReservedNodeOfferingsRequest request = new DescribeReserved
NodeOfferingsRequest();
        DescribeReservedNodeOfferingsResult result = client.describeReserved
NodeOfferings(request);
        Integer count = 0;

        System.out.println("\nFinding nodes to purchase.");
       for (ReservedNodeOffering offering : result.getReservedNodeOfferings())

        {
            if (offering.getNodeType().equals(nodeTypeToPurchase)){

                if (offering.getFixedPrice() < fixedPriceLimit) {
                    matchingNodes.add(offering);
                    printOfferingDetails(offering);
                    count +=1;
                }
            }
        }
        if (count == 0) {
            System.out.println("\nNo reserved node offering matches found.");
        } else {
            System.out.println("\nFound " + count + " matches.");
        }
```

```
    }

    private static void purchaseReservedNodeOffer() throws IOException {
        if (matchingNodes.size() == 0) {
            return;
        } else {
            System.out.println("\nPurchasing nodes.");

            for (ReservedNodeOffering offering : matchingNodes) {
                printOfferingDetails(offering);
              System.out.println("WARNING: purchasing this offering will incur
 costs.");
                System.out.println("Purchase this offering [Y or N]?");
                DataInput in = new DataInputStream(System.in);
                String purchaseOpt = in.readLine();
                if (purchaseOpt.equalsIgnoreCase("y")){

                    try {
                        PurchaseReservedNodeOfferingRequest request = new Pur
chaseReservedNodeOfferingRequest()
                            .withReservedNodeOfferingId(offering.getReserved
NodeOfferingId());
                        ReservedNode reservedNode = client.purchaseReserved
NodeOffering(request);
                        printReservedNodeDetails(reservedNode);
                    }
                    catch (ReservedNodeAlreadyExistsException ex1){
                    }
                    catch (ReservedNodeOfferingNotFoundException ex2){
                    }
                    catch (ReservedNodeQuotaExceededException ex3){
                    }
                    catch (Exception ex4){
                    }
                }
            }
            System.out.println("Finished.");

        }
    }

    private static void printOfferingDetails(
            ReservedNodeOffering offering) {
        System.out.println("\nOffering Match:");
        System.out.format("Id: %s\n", offering.getReservedNodeOfferingId());
        System.out.format("Node Type: %s\n", offering.getNodeType());
        System.out.format("Fixed Price: %s\n", offering.getFixedPrice());
        System.out.format("Offering Type: %s\n", offering.getOfferingType());
        System.out.format("Duration: %s\n", offering.getDuration());
    }

    private static void printReservedNodeDetails(ReservedNode node) {
        System.out.println("\nPurchased Node Details:");
        System.out.format("Id: %s\n", node.getReservedNodeOfferingId());
        System.out.format("State: %s\n", node.getState());
        System.out.format("Node Type: %s\n", node.getNodeType());
        System.out.format("Start Time: %s\n", node.getStartTime());
        System.out.format("Fixed Price: %s\n", node.getFixedPrice());
```

```
        System.out.format("Offering Type: %s\n", node.getOfferingType());
        System.out.format("Duration: %s\n", node.getDuration());

    }
}
```

Purchasing a Reserved Node Offering Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to purchase reserved node offerings.

- purchase-reserved-node-offering
- describe-reserved-node-offerings
- describe-orderable-cluster-options

You can use the following Amazon Redshift APIs to purchase reserved node offerings.

- PurchaseReservedNodeOffering
- DescribeReservedNodeOfferings
- DescribeOrderableClusterOptions

Controlling Access to Amazon Redshift Resources

You can use AWS Identity and Access Management (IAM) to control which AWS users can create, configure, or delete Amazon Redshift clusters and other resources.

Topics

- Overview of Access to Amazon Redshift Resources (p. 115)
- IAM Policy Elements for Amazon Redshift (p. 117)
- Constructing an Amazon Redshift ARN (p. 120)
- Example Policies for Amazon Redshift (p. 121)

Overview of Access to Amazon Redshift Resources

Access to Amazon Redshift resources is controlled at three levels:

- **Cluster management –** The ability to create, configure, and delete clusters is controlled by the permissions given to the user or account associated with your AWS security credentials. AWS users with the proper permissions can use the AWS Management Console, AWS Command Line Interface (CLI), or Amazon Redshift Application Programming Interface (API) to manage their clusters. This section discusses how to control this access by using IAM policies.
- **Cluster connectivity –** Amazon Redshift security groups specify the AWS instances that are authorized to connect to an Amazon Redshift cluster in Classless Inter-Domain Routing (CIDR) format. For information about creating Amazon Redshift, Amazon EC2, and Amazon VPC security groups and associating them with clusters, see Amazon Redshift Cluster Security Groups (p. 43).
- **Database access –** The ability to access database objects, such as tables and views, is controlled by user accounts in the Amazon Redshift database. Users can only access resources in the database that their user accounts have been granted permission to access. You create these Amazon Redshift user accounts and managing permissions by using the CREATE USER, CREATE GROUP, GRANT, and REVOKE SQL statements. For more information, go to Managing Database Security.

Using IAM Users, Groups, and Policies for Cluster Management

Amazon Redshift uses AWS Identity and Access Management (IAM) to control which users in your AWS account can create, modify, or delete clusters for your AWS account. If you do not create IAM users and groups within an account, you must provide the account root credentials to any people who need to manage Amazon Redshift clusters owned by the account. The root credentials provide unrestricted access to all AWS resources owned by the account. As a result, we strongly recommend that instead of using the account credentials to create or manage Amazon Redshift resources, you use IAM. For information about getting AWS security credentials, go to AWS Security Credentials.

Use IAM to do the following:

- Create users and groups under your AWS account.
- Share AWS account resources with the users in the same account.
- Define IAM policies that control the Amazon Redshift actions and resources available to each IAM user or group. By default, a new IAM user has no permissions. You must grant IAM users all permissions they require.
- Configure each user to have his or her own security credentials.

To learn about IAM, go to:

- AWS Identity and Access Management (IAM)
- IAM Getting Started Guide
- Using IAM

AWS Managed Policies for Amazon Redshift

Amazon Redshift offers functionality from other services, such as Amazon CloudWatch metrics that you can use to monitor your cluster performance and set alarms. Functions that are supplied by other services require explicit permissions of their own; access to Amazon Redshift actions does not give access to actions in other services. The AWS IAM console provides the following predefined Amazon Redshift AWS managed policies that you can use to allow the necessary access:

- **Amazon Redshift Read Only Access –** Provides read-only access to Amazon Redshift by using the AWS Management Console.
- **Amazon Redshift Full Access –** Provides full access to Amazon Redshift by using the AWS Management Console.

Alternately, see Example Policies for Amazon Redshift (p. 121) for examples on how to configure explicit permissions for other services that you need to work with Amazon Redshift.

Creating an IAM Group and Users

This section describes how to create an IAM group, attach a policy to the group, and add users to the group. Alternatively, you can attach the policies to the users rather than the group. The following example uses the **Administrator Access** AWS managed policy, but you can use one of the Amazon Redshift AWS managed policies instead. To learn more about IAM groups, users, and policies, go to the IAM documentation listed preceding.

To create the Administrators group

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups**, then click **Create New Group**.
3. In the **Group Name** box, type **`Administrators`** and then click **Next Step**.
4. In the list of policies, select the check box next to the **AdministratorAccess** policy. You can use the **Filter** menu and the **Search** box to filter the list of policies.
5. Click **Next Step**, then click **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user for yourself, add the user to the Administrators group, and create a password for the user

1. In the navigation pane, click **Users** and then click **Create New Users**.
2. In box **1**, enter a user name. Clear the check box next to **Generate an access key for each user**, then click **Create**.
3. In the list of users, click the name (not the check box) of the user you just created. You can use the **Search** box to search for the user name.
4. In the **Groups** section, click **Add User to Groups**.
5. Select the check box next to the **Administrators** group, then click **Add to Groups**.
6. Scroll down to the **Security Credentials** section. Under **Sign-In Credentials**, click **Manage Password**.
7. Select **Assign a custom password**, then enter a password in the **Password** and **Confirm Password** boxes. When you are finished, click **Apply**.

IAM Policy Elements for Amazon Redshift

You create IAM policies to specify which Amazon Redshift actions and resources each IAM user or group has permission to perform. A policy is a JSON document that contains one or more statements; each statement contains elements defining the permissions authorized by the statement.

Within an IAM policy statement, Amazon Redshift supports the following IAM policy elements:

- `Action`, which specifies the action to which the policy applies.
- `Effect`, which specifies whether the action is allowed or denied.
- `Resource`, which specifies the resource or resources to which the policy applies.

For more information about IAM policy elements, go to IAM Policy Elements Reference in *Using IAM*.

The following example shows a policy that allows a user or group to create snapshots from any cluster whose name starts with **my-cluster**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:CreateClusterSnapshot"
      ],
      "Effect": "Allow",
```

```
      "Resource": [
           "arn:aws:redshift:us-east-1:123456789012:snapshot:my-cluster*"
           ]
    }
  ]
}
```

IAM Policy Actions for Amazon Redshift

The `Action` element in an IAM policy defines what a user can do in Amazon Redshift, such as creating clusters, deleting security groups, restoring clusters from snapshots, and so on. When you allow or deny permission to do something in an `Action` element, the permission applies to that action in all of the management interfaces for Amazon Redshift: the Amazon Redshift console, API, CLI, and SDK. For example, setting the `Effect` element to `Allow` for the `redshift:CreateCluster` policy action authorizes a user to create an Amazon Redshift cluster from the launch cluster workflow in the Amazon Redshift console, from the `CreateCluster` API action, from the `create-cluster` CLI command, or from any of the supported AWS SDKs.

Specify the IAM action for Amazon Redshift as `redshift:`*`ActionName`*, where *`ActionName`* is the name of the Amazon Redshift API action. The following example shows an IAM action for the `CreateCluster` API in Amazon Redshift.

```
redshift:CreateCluster
```

For a complete list of Amazon Redshift API actions, go to Actions in the *Amazon Redshift API Reference*.

Redshift also supports the following actions that are not based on the Redshift API:

- The `redshift:ViewQueriesInConsole` action controls whether a user can see queries in the Amazon Redshift console in the **Queries** tab of the **Cluster** section.
- The `redshift:CancelQuerySession` action controls whether a user can terminate running queries and loads from the **Cluster** section in the Amazon Redshift console.

IAM Policy Resources for Amazon Redshift

The `Resource` element in an IAM policy specifies on or with which resources a user can perform the given actions. You can specify one resource or a set of resources by using an Amazon Resource Name (ARN) that includes the name of the service that the resource belongs to (`redshift`), the region (such as `us-east-1`), the account number, the type of resource (such as a cluster or parameter group), and the name of the resource. For reference information on the IAM policy `Resource` element, go to IAM Policy Elements Reference in *Using IAM*.

For information about the format of Amazon Redshift ARNs and examples, see Constructing an Amazon Redshift ARN (p. 120).

For Amazon Redshift snapshots, the name includes the name of both the snapshot and the cluster it was created from: *`<cluster name>`*/*`<snapshot name>`*. If an IAM policy has a `Resource` element with a snapshot ARN, and the cluster node is anything other than *, then all users with that policy must specify the source cluster name for the snapshot when performing certain actions:

- Authorizing cluster snapshot access.
- Copying a cluster snapshot.
- Deleting a cluster snapshot.

- Restoring a cluster from a snapshot.
- Revoking cluster snapshot access.

Some Amazon Redshift actions, such as copying a cluster snapshot and restoring a cluster from a snapshot, reference more than one Amazon Redshift resource. To successfully complete those actions, an IAM user must have policies with `Resource` elements that cover both resources. For example, to restore a cluster from snapshot, an IAM user must have permissions that allow access to the snapshot and that allow creating a cluster with the cluster identifier specified.

Amazon Redshift supports the use of wildcards such as "*" in IAM `Resource` elements. One example is specifying "Resource": "*" to allow operations on all Amazon Redshift resources. Another example is using this ARN to specify all snapshots from the cluster named my-cluster:

```
arn:aws:redshift:us-east-1:123456789012:snapshot:my-cluster/*
```

Amazon Redshift does not support `Resource` elements on all actions. The actions that do support `Resource` elements are:

- AuthorizeClusterSecurityGroup
- AuthorizeSnapshotAccess
- CopyClusterSnapshot
- CreateCluster
- CreateClusterParameterGroup
- CreateClusterSecurityGroup
- CreateClusterSnapshot
- CreateClusterSubnetGroup
- CreateHsmClientCertificate
- CreateHsmConfiguration
- CreateTags
- DeleteCluster
- DeleteClusterParameterGroup
- DeleteClusterSecurityGroup
- DeleteClusterSnapshot
- DeleteClusterSubnetGroup
- DeleteHsmClientCertificate
- DeleteHsmConfiguration
- DeleteTags
- DescribeClusterParameters
- DescribeResize
- DescribeTags
- ModifyCluster
- ModifyClusterParameterGroup
- ModifyClusterSubnetGroup
- RebootCluster
- ResetClusterParameterGroup
- RestoreFromClusterSnapshot
- RevokeClusterSecurityGroupIngress
- RevokeSnapshotAccess

Constructing an Amazon Redshift ARN

You can identify resources that are created in Amazon Web Services by a unique identifier called an Amazon Resource Name (ARN). When using the `Resource` element in an IAM policy to control which Amazon Redshift resources a user or group can access, you specify one or more ARNs that define the resources the users are allowed to access.

An ARN for an Amazon Redshift resource uses the following syntax

```
arn:aws:redshift:<region>:<account_number>:<resource_type>:<name>
```

where:

- *<region>* is the AWS region where the Amazon Redshift resource was created, such as `us-west-2`. For a list of the Amazon Redshift regions, go to Amazon Redshift Regions and Endpoints.
- *<account_number>* is your AWS account number with dashes omitted. To find your account number, sign in to your AWS account at http://aws.amazon.com, click **My Account/Console**, and then click **My Account**.
- *resource_type* is the type of Amazon Redshift resource.
- *<name>* is the resource identifier for the Amazon Redshift resource. For snapshots, the name is a combination of the identifier of the cluster the snapshot was created from and the name of a specific snapshot: *<cluster name>*/*<snapshot name>*.

The following table shows the format you should use when constructing an ARN for a particular Amazon Redshift resource.

Resource Type	ARN Format
Cluster	arn:aws:redshift:*<region>*:*<account_number>*:cluster:*<cluster_name>*
Cluster security group	arn:aws:redshift:*<region>*:*<account_number>*:securitygroup:*<security_group_name>*
CIDR/IP	arn:aws:redshift:*<region>*:*<account_number>*:securitygroupingress:*<security_group_name>*/cidrip/*<IP_range>*
EC2 security group	arn:aws:redshift:*<region>*:*<account_number>*:securitygroupingress:*<security_group_name>*/ec2securitygroup/*<owner>*/*<EC2_security_group_ID>*
HSM client certificate	arn:aws:redshift:*<region>*:*<account_number>*:hsmclientcert:*<HSM_client_certificate_ID>*
HSM configuration	arn:aws:redshift:*<region>*:*<account_number>*:hsmconfiguration:*<HSM_configuration_ID>*
Parameter group	arn:aws:redshift:*<region>*:*<account_number>*:parametergroup:*<parameter_group_name>*
Snapshot	arn:aws:redshift:*<region>*:*<account_number>*:snapshot:*<cluster_name>*/*<snapshot_name>*
Subnet group	arn:aws:redshift:*<region>*:*<account_number>*:subnetgroup:*<subnet_group_name>*

The following table shows examples of ARNs for Amazon Redshift resources that have an AWS account of 123456789012, that were created in the US East (Northern Virginia) region, and that have a resource name that begins with "my":

Resource Type	Sample ARN
Cluster	arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster
Cluster security group	arn:aws:redshift:us-east-1:123456789012:securitygroup:my-security-group
CIDR/IP	arn:aws:redshift:us-east-1:123456789012:securitygroupingress:my-security-group/cidrip/10.0.0.0/16
EC2 security group	arn:aws:redshift:us-east-1:123456789012:securitygroupingress:my-security-group/ec2securitygroup/234567890123/sg-123a4567
HSM client certificate	arn:aws:redshift:us-east-1:123456789012:hsmclientcert:my-hsm-cert
HSM configuration	arn:aws:redshift:us-east-1:123456789012:hsmconfiguration:my-hsm-configuration
Parameter group	arn:aws:redshift:us-east-1:123456789012:parametergroup:my-param-grp
Snapshot	arn:aws:redshift:us-east-1:123456789012:snapshot:myCluster/my-snapshot20130807
Subnet group	arn:aws:redshift:us-east-1:123456789012:subnetgroup:my-subnet-10

Following are examples of ARNs for different sets of snapshots:

- All snapshots created from the cluster `my-cluster`:

```
arn:aws:redshift:us-east-1:123456789012:snapshot:my-cluster/*
```

- All snapshots whose name starts with `my-snapshot`:

```
arn:aws:redshift:us-east-1:123456789012:snapshot:*/my-snapshot*
```

Example Policies for Amazon Redshift

This section provides example IAM policies that illustrate how to grant a user permission to perform specific Amazon Redshift actions, and how to specify the resources those actions can be used on. You can attach these policies to a user or group for which you want to grant access permissions.

Example 1: Allow an IAM user full access to all Amazon Redshift actions and resources

The following policy allows access to all Amazon Redshift actions on all resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

The value `redshift:*` in the `Action` element indicates all the actions in Amazon Redshift.

Example 2: Deny a set of Amazon Redshift actions to an IAM user

By default, all permissions are denied. However, sometimes you need to explicitly deny access to a specific action or set of actions. The following policy allows access to all the Amazon Redshift actions and explicitly denies access to any Amazon Redshift action whose name starts with `Delete`. This policy applies to all Redshift resources in the US East (Northern Virginia) region, that is `us-east-1`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:redshift:us-east-1:*"
    },
   {
     "Action": [
        "redshift:Delete*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:redshift:us-east-1:*"
   }
  ]
}
```

Example 3: Manage clusters

The following policy allows a user to create, delete, modify, and reboot all clusters, and then denies permission to delete any clusters whose name starts with `protected`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:CreateCluster",
        "redshift:DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "redshift:DeleteCluster"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:0123456789012:cluster:protected*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

Example 4: Allow a user to authorize and revoke snapshot access

The following policy allows a user, for example UserA, to do the following:

- Authorize access to any snapshot created from a cluster named `shared`.
- Revoke snapshot access for any snapshot created from the `shared` cluster where the snapshot name starts with `revokable`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:0123456789012:shared/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "redshift:RevokeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:0123456789012:snapshot:*/revokable*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

If UserA has allowed UserB to access a snapshot, UserB must have a policy such as the following to allow UserB to restore a cluster from the snapshot. The following policy allows UserB to describe and restore from snapshots, and to create clusters. The name of these clusters must start with `from-other-account`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:DescribeClusterSnapshots"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "redshift:RestoreFromClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:0123456789012:snapshot:*/*",
       "arn:aws:redshift:us-east-1:2109876543210:cluster:from-other-account*"
```

```
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example 5: Allow a user permissions to copy a cluster snapshot and restore a cluster from a snapshot

The following policy allows a user to copy any snapshot created from the cluster named `big-cluster-1`, and restore any snapshot whose name starts with `snapshot-for-restore`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "redshift:CopyClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:0123456789012:snapshot:big-cluster-1/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "redshift:RestoreFromClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:0123456789012:snapshot:*/snapshot-for-re
store*",
        "arn:aws:redshift:us-east-1:0123456789012:cluster:*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example 6: Grant permission to Amazon Redshift and to common actions and resources for related AWS services

The following example policy allows access to all actions and resources for Amazon Redshift, Amazon SNS, and Amazon CloudWatch, and allows specified actions on all related Amazon EC2 resources under the account.

Note
Resource-level permissions are not supported for the Amazon EC2 actions that are specified in this example policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
        "Action": [
          "sns:*"
        ],
        "Resource": [
          "*"
        ]
      },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
      ],
      "Resource": [
        "*"
      ]
```

```
    }
  ]
}
```

Accessing Amazon Redshift Clusters and Databases

There are several management tools and interfaces you can use to create, manage, and delete Amazon Redshift clusters and the databases within the clusters.

- You work with Amazon Web Services management tools and interfaces to create, manage, and delete Amazon Redshift clusters. These tools and interfaces manage the work of setting up, operating, and scaling a data warehouse; provisioning capacity, monitoring, and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine.
  - You can use the AWS Management Console to interactively create, manage, and delete clusters. The topics in this guide include instructions for using the AWS Management Console to perform specific tasks.
  - You can use one of several AWS management interfaces or SDKs to programmatically create, manage, and delete clusters. For more information, see Using the Amazon Redshift Management Interfaces (p. 128).
- After creating an Amazon Redshift cluster, you can create, manage, and delete databases in the cluster by using client applications or tools that execute SQL statements through the PostgreSQL ODBC or JDBC drivers.
  - For information about installing client SQL tools and connecting to a cluster, see Connecting to a Cluster (p. 138).
  - For information about designing databases and the SQL statements supported by Amazon Redshift, go to the Amazon Redshift Database Developer Guide.

The interfaces used to work with Amazon Redshift clusters and databases comply with the mechanisms that control access, such as security groups and IAM policies. For more information, see Controlling Access to Amazon Redshift Resources (p. 115).

Using the Amazon Redshift Management Interfaces

Topics

- Using the AWS SDK for Java with Amazon Redshift (p. 129)

- Signing an HTTP Request (p. 131)
- Setting Up the Amazon Redshift CLI (p. 134)

Amazon Redshift supports several management interfaces that you can use to use to create, manage, and delete Amazon Redshift clusters; the AWS SDKs, the AWS command line interface, and the Amazon Redshift management API.

Amazon Redshift QUERY API — is a Amazon Redshift management API you can call by submitting a Query request. Query requests are HTTP or HTTPS requests that use the HTTP verbs `GET` or `POST` with a query parameter named *`Action`*. Calling the Query API is the most direct way to access the Amazon Redshift service, but requires that your application handle low-level details such as error handling and generating a hash to sign the request.

- For information about building and signing a Query API request, see Signing an HTTP Request (p. 131).
- For information about the Query API actions and data types for Amazon Redshift, go to the Amazon Redshift API Reference.

AWS SDKs — Amazon Web Services provides Software Development Kits (SDKs) that you can use to perform Amazon Redshift cluster-related operations. Several of the SDK libraries wrap the underlying Amazon Redshift Query API. They integrate the API functionality into the specific programming language and handle many of the low-level details, such as calculating signatures, handling request retries, and error handling. Calling the wrapper functions in the SDK libraries can greatly simplify the process of writing an application to manage an Amazon Redshift cluster.

- Amazon Redshift is supported by the AWS SDKs for Java, .NET, PHP, Python, Ruby, and Node.js. The wrapper functions for Amazon Redshift are documented in the reference manual for each SDK. For a list of the AWS SDKs and links to their documentation, go to Tools for Amazon Web Services.
- This guide provides examples of working with Amazon Redshift using the Java SDK. For more general AWS SDK code examples, go to Sample Code & Libraries.

AWS Command Line Interface (CLI) — provides a set of command line tools that can be used to manage AWS services from Windows, Mac, and Linux computers. The AWS CLI includes commands based on the Amazon Redshift Query API actions.

- For information about installing and setting up the Amazon Redshift CLI, see Setting Up the Amazon Redshift CLI (p. 134).
- For reference material on the Amazon Redshift CLI commands, go to Amazon Redshift in the AWS CLI Reference.

Using the AWS SDK for Java with Amazon Redshift

Topics

- Running Java Examples for Amazon Redshift Using Eclipse (p. 130)
- Running Java Examples for Amazon Redshift from the Command Line (p. 130)
- Setting the Endpoint (p. 131)

The AWS SDK for Java provides a class named `AmazonRedshiftClient`, which you can use to interact with Amazon Redshift. For information about downloading the AWS SDK for Java, go to AWS SDK for Java.

> **Note**
> The AWS SDK for Java provides thread-safe clients for accessing Amazon Redshift. As a best practice, your applications should create one client and reuse the client between threads.

The `AmazonRedshiftClient` class defines methods that map to underlying Amazon Redshift Query API actions. (These actions are described in the Amazon Redshift API Reference). When you call a method, you must create a corresponding request object and response object. The request object includes information that you must pass with the actual request. The response object include information returned from Amazon Redshift in response to the request.

For example, the `AmazonRedshiftClient` class provides the `createCluster` method to provision a cluster. This method maps to the underlying CreateCluster API action. You create a `CreateClusterRequest` object to pass information with the `createCluster` method.

```
AmazonRedshiftClient client = new AmazonRedshiftClient(credentials);
client.setEndpoint("https://redshift.us-east-1.amazonaws.com/");

CreateClusterRequest request = new CreateClusterRequest()
    .withClusterIdentifier("exampleclusterusingjava")
    .withMasterUsername("masteruser")
    .withMasterUserPassword("12345678Aa")
    .withNodeType("dw.hs1.xlarge")
    .withNumberOfNodes(2);

Cluster createResponse = client.createCluster(request);
System.out.println("Created cluster " + createResponse.getClusterIdentifier());
```

Running Java Examples for Amazon Redshift Using Eclipse

General Process of Running Java Code Examples Using Eclipse

1. Create a new **AWS Java Project** in Eclipse.

   Follow the steps in Setting Up the AWS Toolkit for Eclipse in the *AWS Toolkit for Eclipse Getting Started Guide*.
2. Copy the sample code from the section of this document that you are reading and paste it into your project as a new Java class file.
3. Run the code.

Running Java Examples for Amazon Redshift from the Command Line

General Process of Running Java Code Examples from the Command Line

1. Set up and test your environment as follows:
   a. Create a directory to work in and in it create `src`, `bin`, and `sdk` subfolders.
   b. Download the AWS SDK for Java and unzip it to the `sdk` subfolder you created. After you unzip the SDK, you should have four subdirectories in the `sdk` folder, including a `lib` and `third-party` folder.
   c. Supply your AWS credentials to the SDK for Java. For more information, go to Providing AWS Credentials in the AWS SDK for Java in the *AWS SDK for Java Developer Guide*.
   d. Ensure that you can run the Java program compiler (`javac`) and the Java application launcher (`java`) from your working directory. You can test by running the following commands:

```
javac -help
java -help
```

2. Put the code that you want to run in a .java file, and save the file in the `src` folder. To illustrate the process, we use the code from Managing Cluster Security Groups Using the AWS SDK for Java (p. 52) so that the file in the `src` directory is `CreateAndModifyClusterSecurityGroup.java`.
3. Compile the code.

```
javac -cp sdk/lib/aws-java-sdk-1.3.18.jar -d bin src\CreateAndModifyCluster
SecurityGroup.java
```

   If you are using a different version of the AWS SDK for Java, adjust the classpath (`-cp`) for your version.
4. Run the code. In the following command, line breaks are added for readability.

```
java -cp "bin;
          sdk/lib/*;
          sdk/third-party/commons-logging-1.1.1/*;
          sdk/third-party/httpcomponents-client-4.1.1/*;
          sdk/third-party/jackson-core-1.8/*"
          CreateAndModifyClusterSecurityGroup
```

   Change the class path separator as needed for your operating system. For example, for Windows, the separator is ";" (as shown), and for Unix, it is ":". Other code examples may require more libraries than are shown in this example, or the version of the AWS SDK you are working with may have different third-party folder names. For these cases, adjust the classpath (`-cp`) as appropriate.

   To run samples in this document, use a version of the AWS SDK that supports Amazon Redshift. To get the latest version of the AWS SDK for Java, go to AWS SDK for Java.

Setting the Endpoint

By default, the AWS SDK for Java uses the endpoint `https://redshift.us-east-1.amazonaws.com/`. You can set the endpoint explicitly with the `client.setEndpoint` method as shown in the following Java code snippet.

```
client = new AmazonRedshiftClient(credentials);
client.setEndpoint("https://redshift.us-east-1.amazonaws.com/");
```

For a list of supported AWS regions where you can provision a cluster, go to the Regions and Endpoints section in the *Amazon Web Services Glossary*.

Signing an HTTP Request

Topics

- Example Signature Calculation (p. 132)

Amazon Redshift requires that every request you send to the management API be authenticated with a signature. This topic explains how to sign your requests.

If you are using one of the AWS Software Development Kits (SDKs) or the AWS Command Line Interface, request signing is handled automatically, and you can skip this section. For more information about using AWS SDKs, see Using the Amazon Redshift Management Interfaces (p. 128). For more information about using the Amazon Redshift Command Line Interface, go to Amazon Redshift Command Line Reference.

To sign a request, you calculate a digital signature by using a cryptographic hash function. A cryptographic hash is a function that returns a unique hash value that is based on the input. The input to the hash

function includes the text of your request and your secret access key. The hash function returns a hash value that you include in the request as your signature. The signature is part of the `Authorization` header of your request.

Note
For API access, you need an access key ID and secret access key. For more information about creating access keys, see How Do I Get Security Credentials? in the *AWS General Reference*.

After Amazon Redshift receives your request, it recalculates the signature by using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, Amazon Redshift processes the request; otherwise, the request is rejected.

Amazon Redshift supports authentication using AWS Signature Version 4. The process for calculating a signature is composed of three tasks. These tasks are illustrated in the example that follows.

- Task 1: Create a Canonical Request

  Rearrange your HTTP request into a canonical form. Using a canonical form is necessary because Amazon Redshift uses the same canonical form to calculate the signature it compares with the one you sent.
- Task 2: Create a String to Sign

  Create a string that you will use as one of the input values to your cryptographic hash function. The string, called the *string to sign*, is a concatenation of the name of the hash algorithm, the request date, a *credential scope* string, and the canonicalized request from the previous task. The *credential scope* string itself is a concatenation of date, region, and service information.
- Task 3: Create a Signature

  Create a signature for your request by using a cryptographic hash function that accepts two input strings, your string to sign and a *derived key*. The derived key is calculated by starting with your secret access key and using the credential scope string to create a series of hash-based message authentication codes (HMAC-SHA256).

Example Signature Calculation

The following example walks you through the details of creating a signature for CreateCluster request. You can use this example as a reference to check your own signature calculation method. Other reference calculations are included in the Signature Version 4 Test Suite of the Amazon Web Services Glossary.

You can use a GET or POST request to send requests to Amazon Redshift. The difference between the two is that for the GET request your parameters are sent as query string parameters. For the POST request they are included in the body of the request. The example below shows a POST request.

The example assumes the following:

- The time stamp of the request is `Fri, 07 Dec 2012 00:00:00 GMT`.
- The endpoint is US East (Northern Virginia) Region, `us-east-1`.

The general request syntax is:

```
https://redshift.us-east-1.amazonaws.com/
   ?Action=CreateCluster
   &ClusterIdentifier=examplecluster
   &MasterUsername=masteruser
   &MasterUserPassword=12345678Aa
   &NumberOfNode=2
```

```
   &NodeType=dw.hs1.xlarge
   &Version=2012-12-01
   &x-amz-algorithm=AWS4-HMAC-SHA256
   &x-amz-credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/redshift/aws4_re
quest
   &x-amz-date=20121207T000000Z
   &x-amz-signedheaders=content-type;host;x-amz-date
```

The canonical form of the request calculated for Task 1: Create a Canonical Request (p. 132) is:

```
POST
/

content-type:application/x-www-form-urlencoded; charset=utf-8
host:redshift.us-east-1.amazonaws.com
x-amz-date:20121207T000000Z

content-type;host;x-amz-date
55141b5d2aff6042ccd9d2af808fdf95ac78255e25b823d2dbd720226de1625d
```

The last line of the canonical request is the hash of the request body. The third line in the canonical request is empty because there are no query parameters for this API.

The string to sign for Task 2: Create a String to Sign (p. 132) is:

```
AWS4-HMAC-SHA256
20121207T000000Z
20121207/us-east-1/redshift/aws4_request
06b6bef4f4f060a5558b60c627cc6c5b5b5a959b9902b5ac2187be80cbac0714
```

The first line of the *string to sign* is the algorithm, the second line is the time stamp, the third line is the *credential scope*, and the last line is a hash of the canonical request from Task 1: Create a Canonical Request (p. 132). The service name to use in the credential scope is `redshift`.

For Task 3: Create a Signature (p. 132), the derived key can be represented as:

```
derived key = HMAC(HMAC(HMAC(HMAC("AWS4" + YourSecretAccessKey,"20121207"),"us-
east-1"),"redshift"),"aws4_request")
```

The derived key is calculated as series of hash functions. Starting from the inner HMAC statement in the formula above, you concatenate the phrase "AWS4" with your secret access key and use this as the key to hash the data "us-east-1". The result of this hash becomes the key for the next hash function.

After you calculate the derived key, you use it in a hash function that accepts two input strings, your string to sign and the derived key. For example, if you use the secret access key `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` and the string to sign given earlier, then the calculated signature is as follows:

```
9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

The final step is to construct the `Authorization` header. For the demonstration access key `AKIAIOSFODNN7EXAMPLE`, the header (with line breaks added for readability) is:

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20121207/us-
east-1/redshift/aws4_request,
SignedHeaders=content-type;host;x-amz-date,
Signature=9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

Setting Up the Amazon Redshift CLI

This section explains how to set up and run the AWS CLI command line tools for use in managing Amazon Redshift. The Amazon Redshift command line tools run on the AWS Command Line Interface (AWS CLI), which in turn uses Python (http://www.python.org/). The AWS CLI can be run on any operating system that supports Python.

Installation Instructions

To begin using the Amazon Redshift command line tools, you first set up the AWS CLI, and then you add configuration files that define the Amazon Redshift CLI options.

If you have already installed and configured the AWS CLI for another AWS service, you can skip this procedure.

To install the AWS Command Line Interface

1. Go to Getting Set Up with the AWS Command Line Interface, and then follow the instructions for installing the AWS CLI.

   For CLI access, you need an access key ID and secret access key. For more information about creating access keys, see How Do I Get Security Credentials? in the *AWS General Reference*.
2. Create a file containing configuration information such as your access keys, default region, and command output format. Then set the `AWS_CONFIG_FILE` environment variable to reference that file. For detailed instructions, go to Configuring the AWS Command Line Interface in the AWS Command Line Interface User Guide.
3. Run a test command to confirm that the AWS CLI interface is working. For example, the following command should display help information for the AWS CLI:

   ```
   aws help
   ```

   The following command should display help information for Amazon Redshift:

   ```
   aws redshift help
   ```

For reference material on the Amazon Redshift CLI commands, go to Amazon Redshift in the AWS CLI Reference.

Getting Started with the AWS Command Line Interface

Topics

- Step 1: Before You Begin (p. 135)
- Step 2: Launch a Cluster (p. 135)
- Step 3: Authorize Inbound Traffic for Cluster Access (p. 136)
- Step 4: Connect to Your Cluster (p. 137)
- Step 5: Create Tables, Upload Data, and Try Example Queries (p. 138)

- Step 6: Delete Your Sample Cluster (p. 138)

To help you get started using the command line interface, this section shows how to perform basic administrative tasks for an Amazon Redshift cluster. These tasks are very similar to those in the Amazon Redshift Getting Started Guide, but they are focused on the command line interface rather than the Amazon Redshift console.

This section walks you through the process of creating a cluster, creating database tables, uploading data, and testing queries. You will use the Amazon Redshift CLI to provision a cluster and to authorize necessary access permissions. You will then use the SQL Workbench client to connect to the cluster and create sample tables, upload sample data, and execute test queries.

Step 1: Before You Begin

If you don't already have an AWS account, you must sign up for one. Then you'll need to set up the Amazon Redshift command line tools. Finally, you'll need to download client tools and drivers in order to connect to your cluster.

Step 1.1: Sign Up for an AWS account

For information about signing up for an AWS user account, go to the Amazon Redshift Getting Started Guide.

Step 1.2: Download and Install the AWS Command Line Interface (CLI)

If you have not installed the AWS Command Line Interface, see Setting Up the Amazon Redshift CLI (p. 134).

Step 1.3: Download the Client Tools and Drivers

You can use any SQL client tools to connect to an Amazon Redshift cluster with PostgreSQL JDBC or ODBC drivers. If you do not currently have such software installed, you can use SQL Workbench, a free cross-platform tool that you can use to query tables in an Amazon Redshift cluster. The examples in this section will use the SQL Workbench client.

To download SQL Workbench and the PostgreSQL drivers, go to the Amazon Redshift Getting Started Guide.

Step 2: Launch a Cluster

Now you're ready to launch a cluster by using the AWS Command Line Interface (CLI).

> **Important**
> The cluster that you're about to launch will be live (and not running in a sandbox). You will incur the standard usage fees for the cluster until you terminate it. For pricing information, go to the Amazon Redshift pricing page.
> If you complete the exercise described here in one sitting and terminate your cluster when you are finished, the total charges will be minimal.

The `create-cluster` command has a large number of parameters. For this exercise, you will use the parameter values that are described in the following table. Before you create a cluster in a production environment, we recommend that you review all the required and optional parameters so that your cluster configuration matches your requirements. For more information, see create-cluster

Parameter Name	Parameter Value for This Exercise
Cluster Identifier	`examplecluster`
Master Username	`masteruser`

Parameter Name	Parameter Value for This Exercise
Master Password	`TopSecret1`
Node Type	`dw1.xlarge` or the node size that you want to use. For more information, see About Clusters and Nodes (p. 6)
Cluster Type	`single-node`

To create your cluster, type the following command:

```
aws redshift create-cluster --cluster-identifier examplecluster --master-username
 masteruser --master-user-password TopSecret1 --node-type dw1.xlarge --cluster-
type single-node
```

The cluster creation process will take several minutes to complete. To check the status, type the following command:

```
aws redshift describe-clusters --cluster-identifier examplecluster
```

The output will look similar to this:

```
{
    "Clusters": [
        {

       ...output omitted...

            "ClusterStatus": "creating",
            "ClusterIdentifier": "examplecluster",

       ...output omitted...

}
```

When the **ClusterStatus** field changes from `creating` to `available`, your cluster is ready for use.

In the next step, you will authorize access so that you can connect to the cluster.

Step 3: Authorize Inbound Traffic for Cluster Access

You must explicitly grant inbound access to your client in order to connect to the cluster. Your client can be an Amazon EC2 instance or an external computer.

When you created a cluster in the previous step, because you did not specify a security group, you associated the default cluster security group with the cluster. The default cluster security group contains no rules to authorize any inbound traffic to the cluster. To access the new cluster, you must add rules for inbound traffic, which are called ingress rules, to the cluster security group.

Ingress Rules for Applications Running on the Internet

If you are accessing your cluster from the Internet, you will need to authorize a Classless Inter-Domain Routing IP (CIDR/IP) address range. For this example, we will use a CIDR/IP rule of 192.0.2.0/24; you will need to modify this range to reflect your actual IP address and netmask.

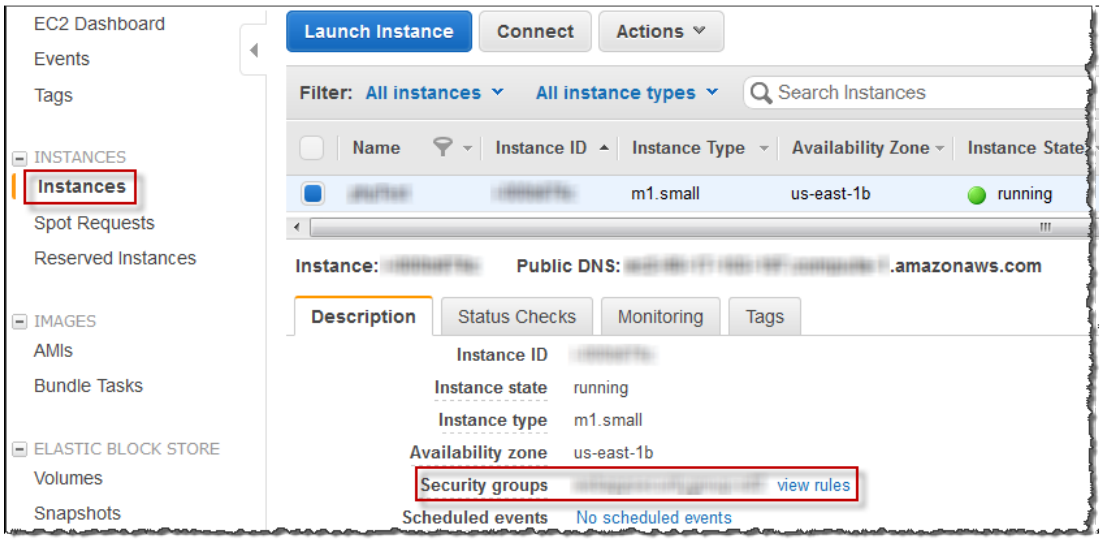
To allow network ingress to your cluster, type the following command:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name default --cidrip 192.0.2.0/24
```

Ingress Rules for EC2 Instances

If you are accessing your cluster from an Amazon EC2 instance, you will need to authorize an Amazon EC2 security group. To do so, you specify the security group name, along with the 12-digit account number of the EC2 security group owner.

You can use the Amazon EC2 console to determine the EC2 security group associated with your instance:



To find your AWS account number, go to http://aws.amazon.com and sign in to the My Account page. Your AWS account number is shown in the upper right-hand corner of that page.

For this example, we will use `myec2securitygroup` for the Amazon EC2 security group name, and `123456789012` for the account number. You will need to modify these to suit your needs.

To allow network ingress to your cluster, type the following command:

```
 aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name default --ec2-security-group-name myec2securitygroup --ec2-security-group-owner 123456789012
```

Step 4: Connect to Your Cluster

Now that you have added an ingress rule to the default cluster security group, incoming connections from a specific CIDR/IP or EC2 Security Group to `examplecluster` are authorized.

You are now ready to connect to the cluster.

For information about connecting to your cluster, go to the Amazon Redshift Getting Started Guide.

Step 5: Create Tables, Upload Data, and Try Example Queries

For information about creating tables, uploading data, and issuing queries, go to the Amazon Redshift Getting Started Guide.

Step 6: Delete Your Sample Cluster

After you have launched a cluster and it is available for use, you are billed for the time the cluster is running, even if you are not actively using it. When you no longer need the cluster, you can delete it.

When you delete a cluster, you must decide whether to create a final snapshot. Because this is an exercise and your test cluster should not have any important data in it, you can skip the final snapshot.

To delete your cluster, type the following command:

```
aws redshift delete-cluster --cluster-identifier examplecluster --skip-final-cluster-snapshot
```

Congratulations! You successfully launched, authorized access to, connected to, and terminated a cluster.

Connecting to a Cluster

You can connect to Amazon Redshift clusters from SQL client tools over Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) connections. Amazon Redshift does not provide or install any SQL client tools or libraries, so you must install them on your client computer or Amazon EC2 instance to use them to work with data in your clusters. You can use most SQL client tools that support JDBC or ODBC drivers.

You can use this section to walk through the process of configuring your client computer or Amazon EC2 instance to use a JDBC or ODBC connection, and related security options for the client connection to the server. Additionally, in this section you can find information about setting up and connecting from two example third-party SQL client tools, SQL Workbench/J and psql, if you don't have a business intelligence tool to use yet. You can also use this section to learn about connecting to your cluster programmatically. Finally, if you encounter issues when attempting to connect to your cluster, you can review the troubleshooting information in this section to identify possible solutions.

Topics

- Configuring Connections in Amazon Redshift (p. 138)
- Connecting to Clusters From Client Tools and Code (p. 162)
- Troubleshooting Connection Issues in Amazon Redshift (p. 173)

Configuring Connections in Amazon Redshift

Use this section to learn how to configure JDBC and ODBC connections to connect to your cluster from SQL client tools. This section describes how to set up JDBC and ODBC connections and how to use Secure Sockets Layer (SSL) and server certificates to encrypt communication between the client and server.

Topics

- JDBC and ODBC Drivers for Amazon Redshift (p. 139)
- Finding Your Cluster Connection String (p. 139)
- Configure a JDBC Connection (p. 140)

- Configure an ODBC Connection (p. 144)
- Configure Security Options for Connections (p. 158)

JDBC and ODBC Drivers for Amazon Redshift

To work with data in your cluster, you need JDBC or ODBC drivers for connectivity from your client computer or instance. Code your applications to use JDBC or ODBC data access APIs, and use SQL client tools that support either JDBC or ODBC.

Amazon Redshift offers JDBC and ODBC drivers for download. Previously, Amazon Redshift recommended PostgreSQL drivers for JDBC and ODBC; if you are currently using those drivers, we recommend moving to the new Amazon Redshift–specific drivers going forward. For more information about how to download the JDBC and ODBC drivers and configure connections to your cluster, see Configure a JDBC Connection (p. 140) and Configure an ODBC Connection (p. 144).

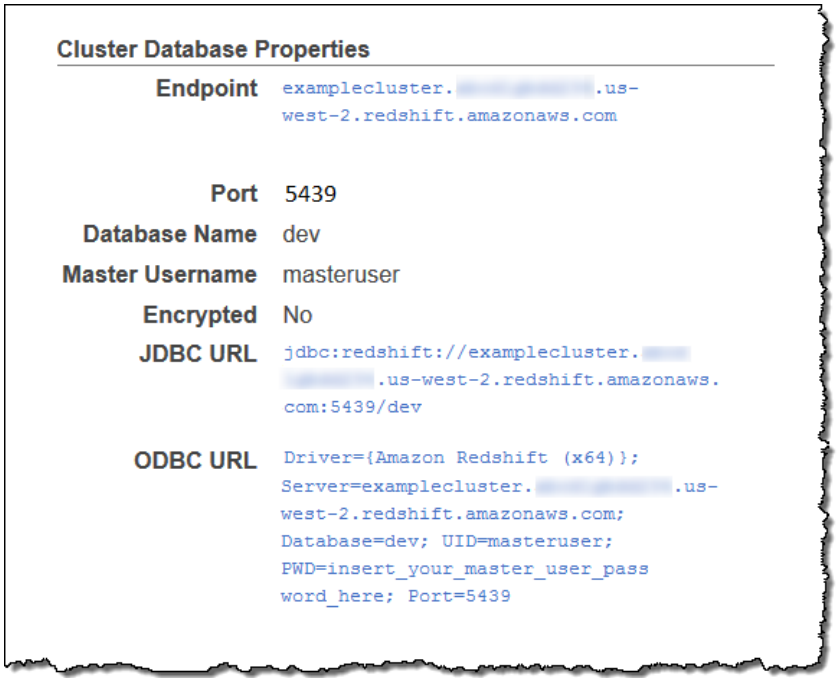
Finding Your Cluster Connection String

To connect to your cluster with your SQL client tool, you need the cluster connection string. You can find the cluster connection string in the Amazon Redshift console, on a cluster's configuration page.

To get your cluster connection string

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. On the **Clusters** page, click the name of the cluster for which you want to get the connection string.
3. On the cluster's **Configuration** tab, under **JDBC URL** or **ODBC URL**, copy the connection string.

   The following example shows the connection strings of a cluster launched in the US West region. If you launched your cluster in a different region, the connection strings will be based that region's endpoint.



Configure a JDBC Connection

You can use a JDBC connection to connect to your Amazon Redshift cluster from many third-party SQL client tools. To do this, you need to download a JDBC driver. Follow the steps in this section if you want to use a JDBC connection.

Topics

- Download the Amazon Redshift JDBC Driver (p. 140)
- Obtain the JDBC URL (p. 140)
- JDBC Driver Configuration Options (p. 142)

Download the Amazon Redshift JDBC Driver

Amazon Redshift offers drivers for tools that are compatible with either the JDBC 4.0 API or the JDBC 4.1 API. For information about the functionality supported by these drivers, go to the Amazon Redshift JDBC Driver Release Notes.

Download one of the following, depending on the version of the JDBC API that your SQL client tool or application uses:

- JDBC 4.1–compatible driver: https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.2.0002.jar

  The class name for this driver is `com.amazon.redshift.jdbc41.Driver`.
- JDBC 4.0–compatible driver: https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.2.0002.jar

  The class name for this driver is `com.amazon.redshift.jdbc4.Driver`.

Then download and review the Amazon Redshift JDBC Driver License Agreement. If you need to distribute these drivers to your customers or other third parties, please email *redshift-pm@amazon.com* to arrange an appropriate license.

Previous JDBC Driver Versions

Download a previous version of the Amazon Redshift JDBC driver only if your tool requires a specific version of the driver. For information about the functionality supported in previous versions of the drivers, go to the Amazon Redshift JDBC Driver Release Notes.

These are previous JDBC 4.1–compatible drivers:

- https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.1.0001.jar
- https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC41-1.1.0.0000.jar

These are previous JDBC 4.0–compatible drivers:

- https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.1.0001.jar
- https://s3.amazonaws.com/redshift-downloads/drivers/RedshiftJDBC4-1.1.0.0000.jar

Obtain the JDBC URL

Before you can connect to your Amazon Redshift cluster from a SQL client tool, you need to know the JDBC URL of your cluster. The JDBC URL has the following format:

jdbc:redshift://*endpoint*:*port*/*database*

Note
A JDBC URL specified with the former format of jdbc:postgresql://*endpoint*:*port*/*database* will still work.

Field	Value
endpoint	The endpoint of the Amazon Redshift cluster.
port	The port number that you specified when you launched the cluster. If you have a firewall, ensure that this port is open for you to use.
database	The database that you created for your cluster.

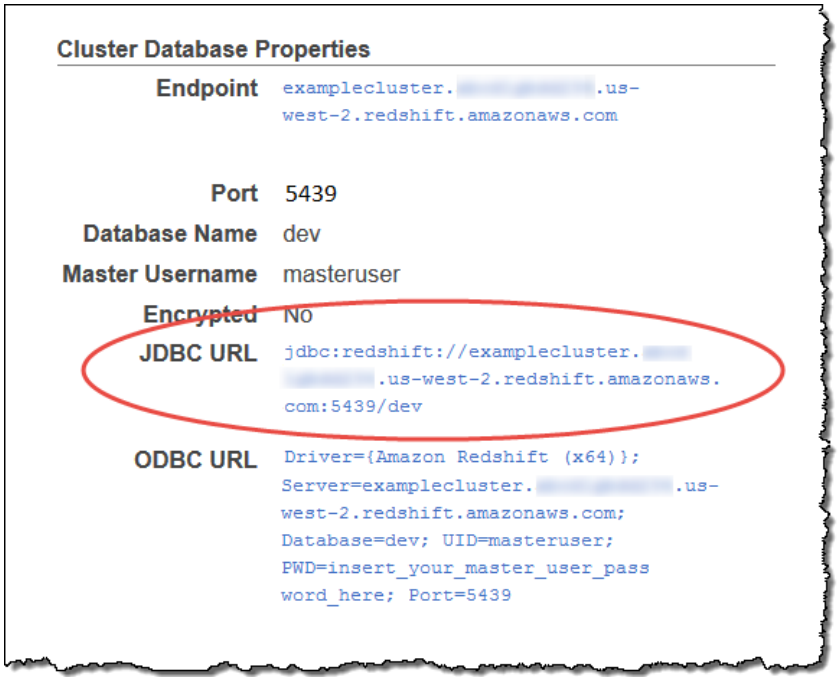
The following is an example JDBC URL:
`jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`

To obtain your JDBC URL

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. At top right, select the region in which you created your cluster.

   If you followed the *Amazon Redshift Getting Started*, select **US West (Oregon)**.
3. In the left navigation pane, click **Clusters**, and then click your cluster.

   If you followed the *Amazon Redshift Getting Started*, click `examplecluster`.
4. On the **Configuration** tab, under **Cluster Database Properties**, copy the JDBC URL of the cluster.



If the client computer fails to connect to the database, you can troubleshoot possible issues. For more information, see Troubleshooting Connection Issues in Amazon Redshift (p. 173).

JDBC Driver Configuration Options

To control the behavior of the Amazon Redshift JDBC driver, you can append the configuration options described in the following table to the JDBC URL .

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
AuthMech	No	DISABLE	The authentication mechanism to use. The following values are possible: • DISABLE—Establish a non-SSL connection. • ALLOW—Attempt to establish a non-SSL connection. If the connection fails, then attempt an SSL connection. • REQUIRE—Establish an SSL connection. • PREFER—Attempt to establish an SSL connection. If the connection fails, then attempt a non-SSL connection.
BlockingRowsMode	No	0	The number of rows to hold in memory. After one row is discarded, another row is loaded in its place.
FilterLevel	No	NOTICE	The minimum severity level of a message that the client will process. The following values are possible, in order from lowest to highest severity: • DEBUG • INFO • NOTICE • WARNING • LOG • ERROR
loginTimeout	Yes	0	The number of seconds to wait before timing out when connecting to the server. If establishing the connection takes longer than this threshold, then the connection is aborted. When this property is set to the default value of 0, connections do not time out.

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
loglevel	Yes		The amount of logging information output by the driver. By default, no logging is performed. Information will be output to the location specified by the LogStream or LogValue option in the driver manager. The following values are possible: • 2 (DEBUG)—Log a lot of detailed information. • 1 (INFO)—Log fewer details. **Note** Only use this property if you are troubleshooting problems with a driver, because it can affect performance.
PWD	Yes		The password to use to connect to the Amazon Redshift server.
socketTimeout	Yes	0	The number of seconds to wait during socket read operations before timing out. If an operation takes longer than this threshold, then the connection is closed. When this property is set to the default value of 0, connections do not time out.
ssl	Yes	false	A value that determines whether to use an SSL connection. The following values are possible: • true—Use SSL. • false—Don't use SSL.
sslFactory	Yes		The factory to use for an SSL connection without a server certificate. The following values are possible: • `com.amazon.redshift.ssl.NonValidatingFactory` • `org.postgresql.ssl.NonValidatingFactory`
tcpKeepAlive	Yes	true	A value that determines whether TCP keepalives are enabled. The following values are possible: • true—Enable TCP keepalives. • false—Disable TCP keepalives.
TCPKeepAliveMinutes	No	5	The threshold for minutes of inactivity before initiating a TCP keepalive transmission.

JDBC Option	Matching PostgreSQL Option Exists?	Default Value	Description
UID	Yes		The user name to use to connect to the Amazon Redshift server.

Configure an ODBC Connection

You can use an ODBC connection to connect to your Amazon Redshift cluster from many third-party SQL client tools and applications. To do this, you need to set up the connection on your client computer or Amazon EC2 instance. If your client tool supports JDBC, you might choose to use that type of connection rather than ODBC due to the ease of configuration that JDBC provides. However, if your client tool doesn't support JDBC, follow the steps in this section to configure an ODBC connection.

Amazon Redshift provides ODBC drivers for Linux, Windows, and Mac OS X operating systems. Before you install an ODBC driver, you need to determine whether your SQL client tool is 32-bit or 64-bit. You should install the ODBC driver that matches the requirements of your SQL client tool; otherwise, the connection will not work. If you use more than one SQL client tool on the same computer or instance, make sure that you download the appropriate drivers. You might need to install both the 32-bit and the 64-bit drivers if the tools differ in their system architecture.

Topics

- Obtain the ODBC URL for Your Cluster (p. 144)
- Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems (p. 146)
- Install the Amazon Redshift ODBC Driver on Linux Operating Systems (p. 148)
- Install the Amazon Redshift ODBC Driver on Mac OS X (p. 150)
- Configure the ODBC Driver on Linux and Mac OS X Operating Systems (p. 151)
- ODBC Driver Configuration Options (p. 155)

Obtain the ODBC URL for Your Cluster

Amazon Redshift displays the ODBC URL for your cluster in the Amazon Redshift console. This URL contains the information that you need to set up the connection between your client computer and the database.

An ODBC URL has the following format:

Driver={*driver*};Server=*endpoint*;Database=*database_name*;UID=*user_name*;PWD=*password*;Port=*port_number*

Field	Value
Driver	The name of the ODBC driver to use. Depending on the driver you download for your architecture, values will be **Amazon Redshift (x86)** (for the 32-bit driver) or **Amazon Redshift (x64)** (for the 64-bit driver).
Server	The endpoint of the Amazon Redshift cluster.
Database	The database that you created for your cluster.

Field	Value
UID	The user name of a user account that has permission to connect to the database. This value is a database permission, not an Amazon Redshift permission, although you can use the master user account that you set up when you launched the cluster.
PWD	The password for the user account to connect to the database.
Port	The port number that you specified when you launched the cluster. If you have a firewall, ensure that this port is open for you to use.

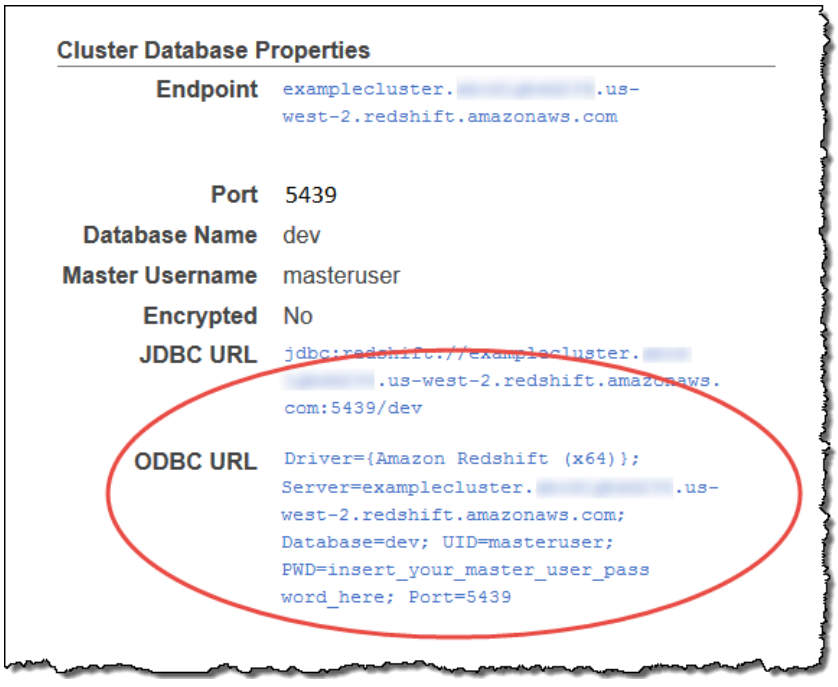
The following is an example ODBC URL: `Driver={Amazon Redshift (x64)}; Server=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com; Database=dev; UID=masteruser; PWD=`*`insert_your_master_user_password_here`*`; Port=5439`

To obtain your ODBC URL

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. At top right, select the region in which you created your cluster.

   If you followed the *Amazon Redshift Getting Started*, select **US West (Oregon)**.
3. In the left navigation pane, click **Clusters**, and then click your cluster.

   If you followed the *Amazon Redshift Getting Started*, click `examplecluster`.
4. On the **Configuration** tab, under **Cluster Database Properties**, copy the ODBC URL of the cluster.



Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems

System Requirements

You install the Amazon Redshift ODBC driver on client computers accessing an Amazon Redshift data warehouse. Each computer where you install the driver must meet the following minimum system requirements:

- Microsoft Windows Vista operating system or later
- 55 MB of available disk space
- Administrator privileges on the client computer
- An Amazon Redshift master user or user account to connect to the database

Installing the Amazon Redshift Driver on Windows Operating Systems

Use the steps in this section to download the Amazon Redshift ODBC drivers for Microsoft Windows operating systems. You should only use a driver other than these if you are running a third-party application that is certified for use with Amazon Redshift and that requires a specific driver for that application.

To install the ODBC driver

1. Download one of the following, depending on the system architecture of your SQL client tool or application:
   - 32-bit: https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC32.msi

     The name for this driver is Amazon Redshift (x86).
   - 64-bit: https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC64.msi

     The name for this driver is Amazon Redshift (x64).

     Note
     Download the MSI package that corresponds to the system architecture of your SQL client tool or application. For example, if your SQL client tool is 64-bit, install the 64-bit driver.

   Then download and review the Amazon Redshift ODBC Driver License Agreement. If you need to distribute these drivers to your customers or other third parties, please email *redshift-pm@amazon.com* to arrange an appropriate license.
2. Double-click the .msi file, and then follow the steps in the wizard to install the driver.

Creating a System DSN Entry for an ODBC Connection on Microsoft Windows

After you download and install the ODBC driver, you need to add a data source name (DSN) entry to the client machine or Amazon EC2 instance. SQL client tools use this data source to connect to the Amazon Redshift database.

To create a system DSN entry

1. In the **Start** menu, in your list of programs, locate the driver folder or folders.

   Note
   If you installed the 32-bit driver, the folder is named **Amazon Redshift ODBC Driver (32-bit)**.
   If you installed the 64-bit driver, the folder is named **Amazon Redshift ODBC Driver (64-bit)**.
   If you installed both drivers, you'll have a folder for each driver.
2. Click **ODBC Administrator**, and then type your administrator credentials if you are prompted to do so.

3. Select the **System DSN** tab if you want to configure the driver for all users on the computer, or the **User DSN** tab if you want to configure the driver for your user account only.
4. Click **Add**. The **Create New Data Source** window opens.
5. Select the **Amazon Redshift** ODBC driver, and then click **Finish**. The **Amazon Redshift ODBC Driver DSN Setup** window opens.
6. Under **Connection Settings**, enter the following information:

   Data Source Name

   Type a name for the data source. You can use any name that you want to identify the data source later when you create the connection to the cluster. For example, if you followed the *Amazon Redshift Getting Started*, you might type `exampleclusterdsn` to make it easy to remember the cluster that you will associate with this DSN.

   Server

   Specify the endpoint for your Amazon Redshift cluster. You can find this information in the Amazon Redshift console on the cluster's details page. For more information, see Configuring Connections in Amazon Redshift (p. 138).

   Port

   Type the port number that the database uses. By default, Amazon Redshift uses 5439, but you should use the port that the cluster was configured to use when it was launched.

   Database

   Type the name of the Amazon Redshift database. If you launched your cluster without specifying a database name, type *`dev`*; otherwise, use the name that you chose during the launch process. If you followed the *Amazon Redshift Getting Started*, type *`dev`*.
7. Under **Credentials**, enter the following information:

   User

   Type the user name for the database user account that you want to use to access the database. If you followed the *Amazon Redshift Getting Started*, type *`masteruser`*.

   Password

   Type the password that corresponds to the database user account.
8. Under **SSL Settings**, specify a value for the following:

   SSL Authentication

   Select a mode for handling Secure Sockets Layer (SSL). In a test environment, you might use `prefer`, but for production environments and when secure data exchange is required, use `verify-ca`. For more information about using SSL, see Connect Using SSL (p. 158).
9. Under **Additional Options**, select one of the following options to specify how to return query results to your SQL client tool or application:
   - **Single Row Mode**. Select this option if you want query results to be returned one row at a time to the SQL client tool or application. Use this option if you plan to query for large result sets and don't want the entire result in memory. Disabling this option improves performance, but it can increase the number of out-of-memory errors.
   - **Use Declare/Fetch**. Select this option if you want query results to be returned to the SQL client tool or application in a specified number of rows at a time. Specify the number of rows in **Cache Size**.
   - **Use Multiple Statements**. Select this option to return results based on multiple SQL statements in a query.

- **Retrieve Entire Result Into Memory**. Select this option if you want query results to be returned all at once to the SQL client tool or application. The default is enabled.

10. In **Logging Options**, specify values for the following:
    - **Log Level**. Select an option to specify whether to enable logging and the level of detail that you want captured in the logs.

      Important
      You should only enable logging when you need to capture information about an issue. Logging decreases performance, and it can consume a large amount of disk space.

    - **Log Path**. Specify the full path to the folder where you want to save log files.

    Then click **OK**.
11. In **Data Type Options**, specify values for the following:
    - **Use Unicode**. Select this option to enable support for Unicode characters. The default is enabled.
    - **Show Boolean Column As String**. Select this option if you want Boolean values to be displayed as string values instead of bit values. If you enable this, **"1"** and **"0"** display instead of **1** and **0**. The default is enabled.
    - **Text as LongVarChar**. Select this option to enable showing text as LongVarChar. The default is enabled.
    - **Max Varchar**. Specify the maximum value for the Varchar data type. A Varchar field with a value larger than the maximum specified will be promoted to LongVarchar. The default value is 255.
    - **Max LongVarChar**. Specify the maximum value for the LongVarChar data type. A LongVarChar field value that is larger than the maximum specified will be truncated. The default value is 8190.
    - **Max Bytea**. Specify the maximum value for the Bytea data type. A Bytea field value that is larger than the maximum specified will be truncated. The default value is 255.

      Note
      The Bytea data type is only used by Amazon Redshift system tables and views, and otherwise is not supported.

    Then click **OK**.
12. Click **Test**. If the client computer can connect to the Amazon Redshift database, you will see the following message: **Connection successful**.

If the client computer fails to connect to the database, you can troubleshoot possible issues. For more information, see Troubleshooting Connection Issues in Amazon Redshift (p. 173).

Install the Amazon Redshift ODBC Driver on Linux Operating Systems

System Requirements

You install the Amazon Redshift ODBC driver on client computers accessing an Amazon Redshift data warehouse. Each computer where you install the driver must meet the following minimum system requirements:

- One of the following Linux distributions (32- and 64-bit editions):
  - Red Hat Enterprise Linux (RHEL) 5.0/6.0
  - CentOS 5.0/6.0
  - SUSE Linux Enterprise Server (SLES) 11
- 75 MB of available disk space
- One of the following ODBC driver managers:
  - iODBC Driver Manager 3.52.7 or later. For more information about the iODBC driver manager and links to download it, go to the Independent Open Database Connectivity website.

- unixODBC 2.3.0 or later. For more information about the unixODBC driver manager and links to download it, go to the unixODBC website.
- An Amazon Redshift master user or user account to connect to the database

Installing the Amazon Redshift Driver on Linux Operating Systems

Use the steps in this section to download and install the Amazon Redshift ODBC drivers on a supported Linux distribution. The installation process will install the driver files in the following directories:

- /opt/amazon/redshiftodbc/lib/32 (for a 32-bit driver)
- /opt/amazon/redshiftodbc/lib/64 (for a 64-bit driver)
- /opt/amazon/redshiftodbc/ErrorMessages
- /opt/amazon/redshiftodbc/Setup

To install the Amazon Redshift ODBC driver

1. Download one of the following, depending on the system architecture of your SQL client tool or application:

   - 32-bit .rpm: https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-32bit-1.1.0.0000-1.i686.rpm
   - 64-bit .rpm: https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC-64bit-1.1.0.0000-1.x86_64.rpm

   The name for both of these drivers is Amazon Redshift ODBC Driver.

   Note
   Download the package that corresponds to the system architecture of your SQL client tool or application. For example, if your client tool is 64-bit, install a 64-bit driver.

   Then download and review the Amazon Redshift ODBC Driver License Agreement. If you need to distribute these drivers to your customers or other third parties, please email *redshift-pm@amazon.com* to arrange an appropriate license.
2. Navigate to the location where you downloaded the package, and then run one of the following commands. Use the command that corresponds to your Linux distribution.

   - On RHEL 5.0/6.0 and CentOS 5.0/6.0 operating systems, run this command:

     ```
     yum --nogpgcheck localinstall RPMFileName
     ```

     Replace *RPMFileName* with the RPM package file name. For example, the following command demonstrates installing the 32-bit driver:

     ```
     yum --nogpgcheck localinstall AmazonRedshiftODBC-32bit-1.x.x.xxxx-x.i686.rpm
     ```

   - On SLES 11, run this command:

     ```
     zypper install RPMFileName
     ```

     Replace *RPMFileName* with the RPM package file name. For example, the following command demonstrates installing the 64-bit driver:

```
zypper install AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.rpm
```

Important
When you have finished installing the drivers, configure them for use on your system. For more information on driver configuration, see Configure the ODBC Driver on Linux and Mac OS X Operating Systems (p. 151).

Install the Amazon Redshift ODBC Driver on Mac OS X

System Requirements

You install the driver on client computers accessing an Amazon Redshift data warehouse. Each computer where you install the driver must meet the following minimum system requirements:

- Mac OS X version 10.6.8 or later
- 215 MB of available disk space
- iODBC Driver Manager version 3.52.7 or later. For more information about the iODBC driver manager and links to download it, go to the Independent Open Database Connectivity website.
- An Amazon Redshift master user or user account to connect to the database

Installing the Amazon Redshift Driver on Mac OS X

Use the steps in this section to download and install the Amazon Redshift ODBC driver on a supported version of Mac OS X. The installation process will install the driver files in the following directories:

- /opt/amazon/redshift/lib/universal
- /opt/amazon/redshift/ErrorMessages
- /opt/amazon/redshift/Setup

To install the Amazon Redshift ODBC driver on Mac OS X

1. Download https://s3.amazonaws.com/redshift-downloads/drivers/AmazonRedshiftODBC.dmg. The name for this driver is Amazon Redshift ODBC Driver.

   Then download and review the Amazon Redshift ODBC Driver License Agreement. If you need to distribute these drivers to your customers or other third parties, please email *redshift-pm@amazon.com* to arrange an appropriate license.
2. Double-click **AmazonRedshiftODBC.dmg** to mount the disk image.
3. Double-click **AmazonRedshiftODBC.pkg** to run the installer.
4. Follow the steps in the installer to complete the driver installation process. You'll need to agree to the terms of the license agreement to perform the installation.

Important
When you have finished installing the driver, configure it for use on your system. For more information on driver configuration, see Configure the ODBC Driver on Linux and Mac OS X Operating Systems (p. 151).

Configure the ODBC Driver on Linux and Mac OS X Operating Systems

On Linux and Mac OS X operating systems, you use an ODBC driver manager to configure the ODBC connection settings. ODBC driver managers use configuration files to define and configure ODBC data sources and drivers. The ODBC driver manager that you use depends on the operating system that you use. For more information about the supported ODBC driver managers to configure the Amazon Redshift ODBC drivers, see System Requirements (p. 148) for Linux operating systems and System Requirements (p. 150) for Mac OS X operating systems.

Three files are required for configuring the Amazon Redshift ODBC driver: `amazon.redshiftodbc.ini`, `odbc.ini`, and `odbcinst.ini`.

If you installed to the default location, the `amazon.redshiftodbc.ini` configuration file is located in one of the following directories:

- /opt/amazon/redshiftodbc/lib/32 (for the 32-bit driver on Linux operating systems)
- /opt/amazon/redshiftodbc/lib/64 (for the 64-bit driver on Linux operating systems)
- /opt/amazon/redshift/lib/universal (for the driver on Mac OS X)

Additionally, under /opt/amazon/redshiftodbc/Setup on Linux or /opt/amazon/redshift/Setup on Mac OS X, there are sample `odbc.ini` and `odbcinst.ini` files for you to use as examples for configuring the Amazon Redshift ODBC driver and the data source name (DSN).

We don't recommend using the Amazon Redshift ODBC driver installation directory for the configuration files. The sample files in the Setup directory are for example purposes only. If you reinstall the Amazon Redshift ODBC driver at a later time, or upgrade to a newer version, the installation directory is overwritten and you'll lose any changes you might have made to those files.

To avoid this, you should copy the amazon.redshiftodbc.ini file to a directory other than the installation directory. If you copy this file to the user's home directory, add a period (.) to the beginning of the file name to make it a hidden file.

For the odbc.ini and odbcinst.ini files, you should either use the configuration files in the user's home directory or create new versions in another directory. By default, your Linux or Mac OS X operating system should have an .odbc.ini file and an .odbcinst.ini file in the user's home directory (/home/$USER or ~/.). These default files are hidden files, which are indicated by the dot (.) in front of the file name, and they will only display when you use the -a flag to list the directory contents.

Whichever option you choose for the odbc.ini and odbcinst.ini files, you will need to modify them to add driver and DSN configuration information. If you chose to create new files, you also need to set environment variables to specify where these configuration files are located.

Configuring the odbc.ini File

You use the odbc.ini file to define data source names (DSNs).

Use the following format on Linux operating systems:

```
[ODBC Data Sources]
dsn_name=driver_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
```

```
Database=database_name
locale=locale
```

The following example shows the configuration for odbc.ini on Linux operating systems:

```
[ODBC Data Sources]
Amazon_Redshift_x32=Amazon Redshift (x86)
Amazon_Redshift_x64=Amazon Redshift (x64)

[Amazon Redshift (x86)]
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US

[Amazon Redshift (x64)]
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

Use the following format on Mac OS X operating systems:

```
[ODBC Data Sources]
dsn_name=driver_name

[dsn_name]
Driver=path/libamazonredshiftodbc.dylib

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

The following example shows the configuration for odbc.ini on Mac OS X operating systems:

```
[ODBC Data Sources]
Amazon_Redshift_dylib=Amazon Redshift DSN for Mac OS X

[Amazon Redshift DSN for Mac OS X]
Driver=/opt/amazon/redshift/lib/universal/libamazonredshiftodbc.dylib
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

Configuring the odbcinst.ini File

You use the odbcinst.ini file to define ODBC drivers.

Use the following format on Linux operating systems:

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

The following example shows the odbcinst.ini configuration for both the 32-bit and 64-bit drivers installed in the default directories on Linux operating systems:

```
[ODBC Drivers]
Amazon Redshift (x86)=Installed
Amazon Redshift (x64)=Installed

[Amazon Redshift (x86)]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so

[Amazon Redshift (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

Use the following format on Mac OS X operating systems:

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/libamazonredshiftodbc.dylib

...
```

The following example shows the odbcinst.ini configuration for the driver installed in the default directory on Mac OS X operating systems:

```
[ODBC Drivers]
Amazon RedshiftODBC DSN=Installed

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for Mac OS X
Driver=/opt/amazon/redshift/lib/universal/libamazonredshiftodbc.dylib
```

Configuring Environment Variables for Driver Configuration Files

In order for the Amazon Redshift ODBC driver to function properly, you need to set a number of environmental variables, as described following.

Set an environment variable to specify the path to the driver manager libraries:

- On Linux, set `LD_LIBRARY_PATH` to point to the directory containing the driver manager libraries. For more information on supported driver managers, see Install the Amazon Redshift ODBC Driver on Linux Operating Systems (p. 148).
- On Mac OS X, set `DYLD_LIBRARY_PATH` to point to the directory containing the driver manager libraries. For more information on supported driver managers, see Install the Amazon Redshift ODBC Driver on Mac OS X (p. 150).

Optionally, set `AMAZONREDSHIFTODBCINI` to point to your amazon.redshiftodbc.ini file. `AMAZONREDSHIFTODBCINI` must specify the full path, including the file name. You must either set this variable, or place this file in a location where the system will find it in a search. The following search order is used to locate the amazon.redshiftodbc.ini file:

1. If the `AMAZONREDSHIFTODBCINI` environment variable is defined, then the driver searches for the file specified by the environment variable.
2. The current working directory of the application is searched for a file named amazon.redshiftodbc.ini not beginning with a period.
3. The directory ~/ (that is, $HOME) is searched for a hidden file named .amazon.redshiftodbc.ini.
4. The directory /etc is searched for a file named amazon.redshiftodbc.ini not beginning with a period.

If you decide to use a directory other than the user's home directory for the odbc.ini and odbcinst.ini files, you also need to set environment variables to specify where the configuration files appear:

- Set `ODBCINI` to point to your odbc.ini file.
- Set `ODBCSYSINI` to point to the directory containing the odbcinst.ini file.

If you are on Linux, your driver manager libraries are located in the /usr/local/lib directory, your odbc.ini and amazon.redshiftodbc.ini files are located in the /etc directory, and your odbcinst.ini file is located in the /usr/local/odbc directory, then set the environment variables as shown in the following example:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export ODBCINI=/etc/odbc.ini
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
export ODBCSYSINI=/usr/local/odbc
```

If you are on Mac OS X, your driver manager libraries are located in the /usr/local/lib directory, your odbc.ini and amazon.redshiftodbc.ini files are located in the /etc directory, and your odbcinst.ini file is located in the /usr/local/odbc directory, then set the environment variables as shown in the following example:

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
export ODBCINI=/etc/odbc.ini
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
export ODBCSYSINI=/usr/local/odbc
```

ODBC Driver Configuration Options

You can use the configuration options described in the following table to control the behavior of the Amazon Redshift ODBC driver.

In Windows, you typically set driver options when you configure a DSN. You can also set driver options in the connection string when you connect programatically, or by adding or changing registry keys in HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*your_DSN*. For more information about configuring a DSN, see Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems (p. 146). For an example of setting driver options in a connection string, see Connect to Your Cluster Programmatically (p. 169).

In Linux and Mac OS X, you set driver configuration options in your odbc.ini and amazon.redshiftodbc.ini files, as described in Configure the ODBC Driver on Linux and Mac OS X Operating Systems (p. 151). Configuration options set in an amazon.redshiftodbc.ini file apply to all connections, whereas configuration options set in an odbc.ini file are specific to a connection. Configuration options set in odbc.ini take precedence over configuration options set in amazon.redshiftodbc.ini.

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
BoolsAsChar	Yes	0	When this option is enabled (1), the driver exposes Boolean values as data type SQL_VARCHAR with a length of 5. When this option is disabled (0), the driver exposes Boolean values as data type SQL_BIT.
Database	Yes		The name of the database to use when the connection is established.
Driver	Yes		The location of the Amazon Redshift ODBC driver shared object file.
Fetch	Yes	100	The number of rows that the driver returns when UseDeclareFetch is enabled.
KeepAlive	No. If keepalives are disabled at the TCP/IP level, KeepAliveTime and KeepAliveInterval are set to 0.	1	When this option is enabled (1), the driver uses TCP keepalives to prevent connections from timing out. When this option is disabled (0), the driver does not use TCP keepalives.
KeepAliveCount	No	0	The number of TCP keepalive packets that can be lost before the connection is considered broken. When this option is set to 0, the driver uses the TCP/IP system default for this setting.

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
KeepAliveTime	Yes	0	The number of seconds of inactivity before the driver sends a TCP keepalives packet. When this option is set to 0, the driver uses the TCP/IP system default for this setting.
KeepAliveInterval	Yes	0	The number of seconds between each TCP keepalive retransmission.
Locale	No	en-US	The locale to use for error messages.
MaxBytea	Yes	255	The maximum data length for BYTEA columns, in bytes.
MaxLongVarChar	Yes	8190	The maximum data length for LONG VARCHAR columns, in UTF-8 code units.
MaxVarchar	Yes	255	The maximum data length for VARCHAR columns, in UTF-8 code units.
Port	Yes		The port to connect to on the Redshift server. **Note** By default, Amazon Redshift uses port 5439.
PWD or Password	Yes		The password to use to connect to the Amazon Redshift server.
Server or Servername	Yes		The IP address or hostname of the Amazon Redshift server.
SingleRowMode	No	0	When this option is enabled (1), the driver returns query results one row at a time. Enable this option if you plan to query large results and don't want to retrieve the entire result into memory. When this option and UseDeclareFetch are both disabled (0), the driver retrieves the entire query result into memory. **Note** If UseDeclareFetch is enabled (1), then it takes precedence over SingleRowMode. If SingleRowMode is enabled (1) and UseDeclareFetch is disabled (0), then SingleRowMode takes precedence over UseMultipleStatements.
SSLMode	Yes	require	The SSL certificate verification mode to use when connecting. For more information about possible SSL modes to use, see Using SSL and Server Certificates in ODBC (p. 159).

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
TextAsLongVarchar	Yes	0	When this option is enabled (1), the driver returns TEXT columns as LONG VARCHAR data. When this option is disabled (0), the driver returns TEXT columns as TEXT data.
UID	Yes		The user name to use to connect to the Amazon Redshift server.
UseDeclareFetch	Yes	0	When this option is enabled (1), the driver returns a specific number of rows at a time. To set the number of rows, use the Fetch option. When this option is disabled (0) and SingleRowMode is enabled (1), the driver returns query results one row at a time. If SingleRowMode is also disabled (0), then the driver retrieves the entire query result into memory. **Note** If UseDeclareFetch is enabled, then UseDeclareFetch takes precedence over SingleRowMode and UseMultipleStatements.
UseMultipleStatements	No	0	When this option is enabled (1), the driver can run queries that are split into separate statements. When this option is disabled (0), the driver runs queries as single statements. **Note** If UseDeclareFetch is enabled (1), then UseDeclareFetch takes precedence over UseMultipleStatements. If UseDeclareFetch is disabled (0) but SingleRowMode is enabled (1), then SingleRowMode takes precedence over UseMultipleStatements.
Username	Yes		The same information as UID (the user name to use to connect to the Amazon Redshift server). If UID is defined, then UID takes precedence over Username.

ODBC Option	Matching PostgreSQL option exists?	Default Value	Description
UseUnicode	No	0	When this option is enabled (1), the driver returns data as Unicode character types: • CHAR is returned as SQL_WCHAR. • VARCHAR is returned as SQL_WVARCHAR. • TEXT is returned as SQL_WLONGVARCHAR. When this option is disabled (0), the driver returns data as regular SQL types: • CHAR is returned as SQL_CHAR. • VARCHAR is returned as SQL_VARCHAR. • TEXT is returned as SQL_LONGVARCHAR.

Configure Security Options for Connections

Amazon Redshift supports Secure Sockets Layer (SSL) connections to encrypt data and server certificates to validate the server certificate that the client connects to.

Topics

- Connect Using SSL (p. 158)
- Using SSL and Server Certificates in ODBC (p. 159)
- Using SSL and Server Certificates in Java (p. 160)

Connect Using SSL

To support SSL connections, Amazon Redshift creates and installs a self-signed SSL certificate on each cluster. The public key is stored at https://s3.amazonaws.com/redshift-downloads/redshift-ssl-ca-cert.pem. When you download this certificate on Windows operating systems or Linux operating systems, the file ends with the .pem extension. When you download this certificate on Mac OS X operating systems, the file ends with the .er extension. The expected MD5 checksum number is 1314113b03bf3e6c49ea0b1d2dc03121. You can use the Md5sum program (on Linux operating systems) or other tool (on Windows and Mac OS X operating systems) to verify that the certificate that you downloaded matches this expected MD5 checksum number.

> **Important**
> SSL support in Amazon Redshift is strictly for encrypting the connection between your client and your cluster; it should not be relied on for authenticating the server. To authenticate the server, install the public key (.pem file) for the SSL certificate on your client and use the key to connect to your clusters.

By default, cluster databases accept a connection whether it uses SSL or not. To configure your cluster to require an SSL connection, set the `require_SSL` parameter to `true` in the parameter group that is associated with the cluster. For information about modifying a parameter group, see Modifying a Parameter Group (p. 74).

Amazon Redshift supports the Elliptic Curve Diffie—Hellman Ephemeral (ECDHE) key agreement protocol. With ECDHE, the client and server each have an elliptic curve public-private key pair that is used to establish a shared secret over an insecure channel. You do not need to configure anything in Amazon Redshift to enable ECDHE; if you connect from a SQL client tool that uses ECDHE to encrypt communication between the client and server, Amazon Redshift will use the provided cipher list to make the appropriate connection. For more information, see Elliptic Curve Diffie—Hellman on Wikipedia and Ciphers on the OpenSSL website.

Using SSL and Server Certificates in ODBC

ODBC DSNs contain an `sslmode` setting that determines how to handle encryption for client connections and server certificate verification. Amazon Redshift supports the following `sslmode` values from the client connection:

- `disable`

  SSL is disabled and the connection is not encrypted.
- `allow`

  SSL is used if the server requires it.
- `prefer`

  SSL is used if the server supports it. Amazon Redshift supports SSL, so SSL is used when you set `sslmode` to `prefer`.
- `require`

  SSL is required.
- `verify-ca`

  SSL must be used and the server certificate must be verified.

Amazon Redshift does not support `verify-full`. For more information about `sslmode` options, see SSL Support in the PostgreSQL documentation.

To determine whether SSL is used and server certificates are verified in a connection between the client and the server, you need to review the `sslmode` setting for your ODBC DSN on the client and the `require_SSL` setting for the Amazon Redshift cluster on the server. The following table describes the encryption result for the various client and server setting combinations:

sslmode (client)	require_SSL (server)	Result
`disable`	`false`	The connection is not encrypted.
`disable`	`true`	The connection cannot be made because the server requires SSL and the client has SSL disabled for the connection.
`allow`	`true`	The connection is encrypted.
`allow`	`false`	The connection is not encrypted.
`prefer` or `require`	`true`	The connection is encrypted.
`prefer` or `require`	`false`	The connection is encrypted.
`verify-ca`	`true`	The connection is encrypted and the server certificate is verified.

sslmode (client)	require_SSL (server)	Result
`verify-ca`	`false`	The connection is encrypted and the server certificate is verified.

Connect Using the Server Certificate with ODBC on Microsoft Windows

If you want to connect to your cluster using SSL and the server certificate, you need to download the certificate to your client computer or Amazon EC2 instance, and then configure the ODBC DSN.

1. Download the Amazon Redshift server certificate to your client computer at `%APPDATA%\Roaming\postgresql\`, and save the file as `root.crt`.
2. Open **ODBC Data Source Administrator**, and add or edit the system DSN entry for your ODBC connection. For **SSL Mode**, select `verify-ca` and then click **Save**.

   For more information about configuring the ODBC DSN, see Configure an ODBC Connection (p. 144).

Connect Without Using the Server Certificate with ODBC

If you want to connect to your cluster without using the Amazon Redshift server certificate, you can configure your ODBC DSN to use one of the following SSL modes: **allow**, **prefer**, or **require**. With these settings, the connection will use SSL but will not verify the server certificate.

Using SSL and Server Certificates in Java

SSL provides one layer of security by encrypting data that moves between your client and cluster. Using a server certificate provides an extra layer of security by validating that the cluster is an Amazon Redshift cluster. It does so by checking the server certificate that is automatically installed on all clusters that you provision. For more information about using server certificates with JDBC, go to Configuring the Client in the PostgreSQL documentation.

Connect Using the Server Certificate in Java

This section explains how to add the Amazon Redshift certificate to a Java keystore. The instructions assume that the Java installation indicated by your `JAVA_HOME` environment variable is used by the client you use to connect to your cluster. Additionally, we recommend that you run the commands in the task as root user.

To connect using a server certificate

Use the keytool program to add the Amazon Redshift certificate to the Java system truststore on your client computer or Amazon EC2 instance. At a command prompt, use the following command:

```
${JAVA_HOME}/bin/keytool -keystore ${JAVA_HOME}/lib/security/cacerts -import -
alias <alias> -file <certificate_filename>
```

Where *<alias>* is any user-provided string value and *<certificate_filename>* is the full path to the certificate file that you downloaded from https://s3.amazonaws.com/redshift-downloads/redshift-ssl-ca-cert.pem.

You will be prompted to enter and re-enter a password that will be used for working with the keystore. The default password for the **cacerts** keystore is **changeit**, although you should use whatever password belongs to the keystore if you have changed it.

If you do not have access to the **cacerts** keystore in the command preceding, you can create your own truststore by using the following command:

```
${JAVA_HOME}/bin/keytool -keystore <keystore_name> -alias <alias> -import -file
 <certificate_filename>
```

Where *<keystore_name>* is the keystore that your client application uses, and *<alias>* and *<certificate_filename>* correspond to the descriptions as previously described. You will be prompted to enter and re-enter a password that will be used for working with the keystore. Make note of this password for later use.

Then, when you start your Java application you must specify this keystore and password to use:

```
java -Djavax.net.ssl.trustStore=keystore_name -Djavax.net.ssl.trustStorePass
word=password com.mycompany.MyApp
```

If you have problems with adding the certificate to your truststore, you can review log information for errors. If you use SQL Workbench/J for the *Amazon Redshift Getting Started* and other related exercises in the Amazon Redshift documentation, go to Configuration Directory in the SQL Workbench/J documentation to find where logs are stored. If you are using a different client tool, refer to the documentation that accompanies your tool to find where logs are stored.

Specify the following properties in your connection string:

- Property name: `ssl`
- Property value: `true`

For example, in SQL Workbench/J, you can specify the connection string with the ssl=true parameter in the JDBC URL:

```
jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazon
aws.com:5439/dev?ssl=true
```

For more information about JDBC connections, see Obtain the JDBC URL (p. 140).

In Java code you can specify the connection string as follows:

```
Connection conn = null;
Properties props = new Properties();
props.setProperty("ssl", "true");
conn = DriverManager.getConnection(<jdbc-connection-string>, props);
```

For a full Java example using SSL, see Connecting to a Cluster by Using Java (p. 169).

Connect Without Using the Server Certificate in Java

If you do not use the server certificate, you can still connect to your Amazon Redshift cluster; however, your client will not be able to validate that it is connecting to an Amazon Redshift cluster. If you don't use the certificate, you can still connect by using SSL.

To connect without using a server certificate

Specify the following properties in your connection string:

- Property name: `sslfactory`
- Property value: `com.amazon.redshift.ssl.NonValidatingFactory`

The following is an example connection string for SQL Workbench/J:

```
jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazon
aws.com:5439/dev?ssl=true&sslfactory=com.amazon.redshift.ssl.NonValidatingFactory
```

For more information about JDBC connections, see Obtain the JDBC URL (p. 140).

In Java code you can specify the connection string as follows:

```
Connection conn = null;
Properties props = new Properties();
props.setProperty("ssl", "true");
props.setProperty("sslfactory", "com.amazon.redshift.ssl.NonValidatingFactory");
conn = DriverManager.getConnection(<jdbc-connection-string>, props);
```

Connecting to Clusters From Client Tools and Code

This section provides some options for third-party tools to connect to the cluster if you do not already have a business intelligence tool to do so. Additionally, it describes how to connect to your cluster programmatically.

Topics

- Connect to Your Cluster by Using SQL Workbench/J (p. 162)
- Connect to Your Cluster by Using the psql Tool (p. 166)
- Connect to Your Cluster Programmatically (p. 169)

Connect to Your Cluster by Using SQL Workbench/J

Amazon Redshift does not provide or install any SQL client tools or libraries, so you must install any that you want to use with your clusters. If you already have a business intelligence application or any other application that can connect to your clusters using a standard PostgreSQL JDBC or ODBC driver, then you can skip this section. If you don't already have an application that can connect to your cluster, this section presents one option for doing so using SQL Workbench/J, a free, DBMS-independent, cross-platform SQL query tool.

Topics

- Install SQL Workbench/J (p. 162)
- Connect to Your Cluster over a JDBC Connection in SQL Workbench/J (p. 163)
- Test the SQL Workbench/J Connection (p. 165)

Install SQL Workbench/J

The *Amazon Redshift Getting Started* uses SQL Workbench/J. In this section, we explain in detail how to connect to your cluster by using SQL Workbench/J.

To install SQL Workbench/J

1. Go to the SQL Workbench/J website and download the appropriate package for your operating system on your client computer or Amazon EC2 instance.

2. Go to the Installing and starting SQL Workbench/J page. Follow the instructions for installing SQL Workbench/J on your system.

   Note
   SQL Workbench/J requires the Java Runtime Environment (JRE) be installed on your system. Ensure you are using the correct version of the JRE required by the SQL Workbench/J client. To determine which version of the Java Runtime Environment is running on your system, do one of the following:

   - Mac: In the **System Preferences**, click the Java icon.
   - Windows: In the **Control Panel**, click the Java icon.
   - Any system: In a command shell, type `java -version`. You can also visit http://www.java.com, click the Do I Have Java? link, and click on the **Verify Java** button.

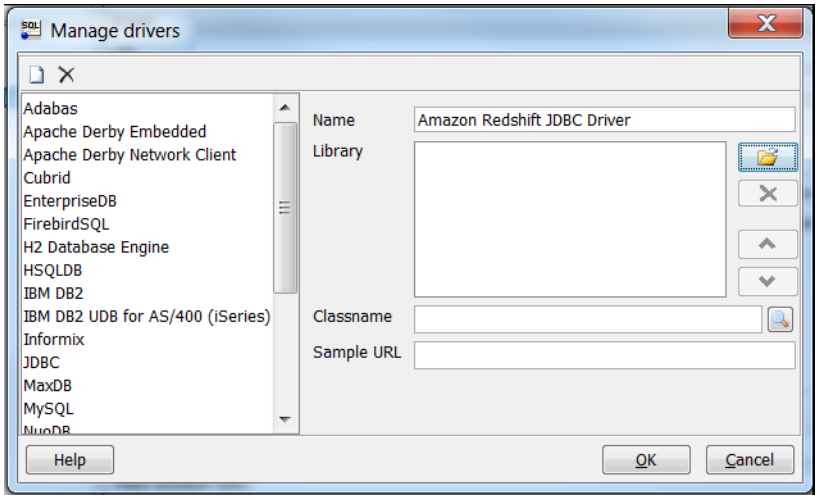
   For information about installing and configuring the Java Runtime Environment, go to http://www.java.com.

Connect to Your Cluster over a JDBC Connection in SQL Workbench/J

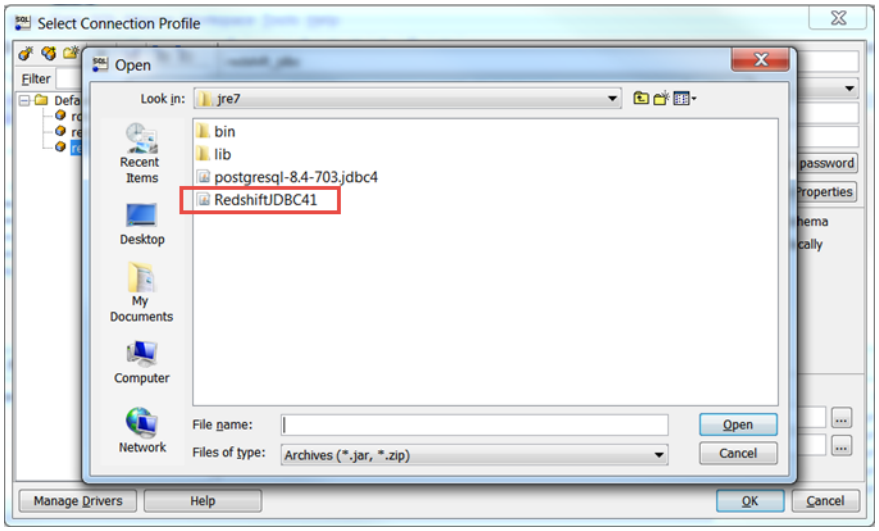
Before you perform the steps in this procedure, make sure that your client computer or Amazon EC2 instance has the recommended Amazon Redshift JDBC driver. For more information, see Configure a JDBC Connection (p. 140).

To use a JDBC connection in SQL Workbench/J

1. Open SQL Workbench/J.
2. Click **File**, and then click **Connect window**.
3. Click **Create a new connection profile**.
4. In the **New profile** box, type a name for the profile. For example, examplecluster_jdbc.
5. Click **Manage Drivers**. The **Manage Drivers** dialog opens. In the **Name** box, type a name for the driver.



   Click the folder icon next to the **Library** box, navigate to the location of the driver, click it, and then click **Open**.

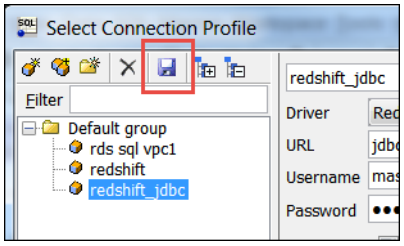


If the **Please select one driver** dialog box displays, select **com.amazon.redshift.jdbc4.Driver** or **com.amazon.redshift.jdbc41.Driver** and click **OK**. SQL Workbench/J automatically completes the **Classname** box. Leave the **Sample URL** box blank, and then click **OK**.

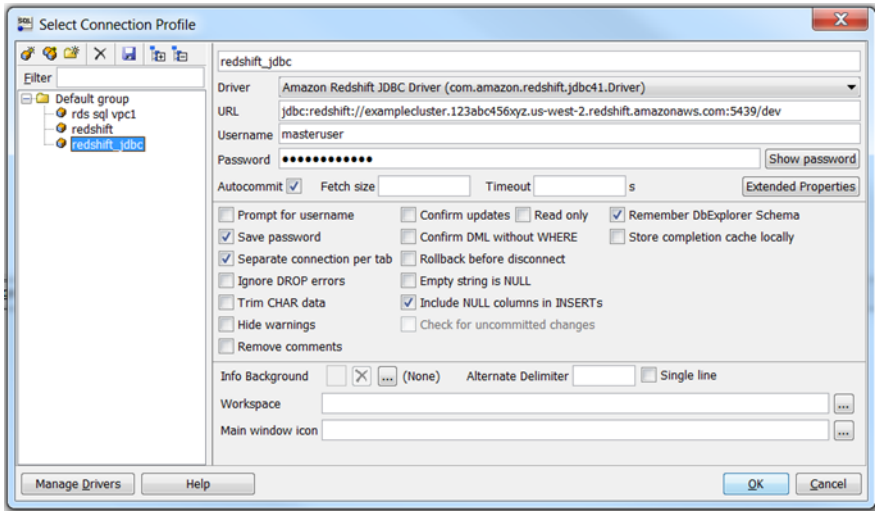
6. In the **Driver** box, select the driver you just added.
7. In **URL**, copy the JDBC URL from the Amazon Redshift console and paste it here.

   For more information about finding the JDBC URL, see Configure a JDBC Connection (p. 140).
8. In **Username**, type the name of the master user.

   If you are following the *Amazon Redshift Getting Started*, type `masteruser`.
9. In **Password**, type the password associated with the master user account.
10. Select the **Autocommit** box.
11. Click the **Save profile list** icon, as shown below:



12. Click **OK**.



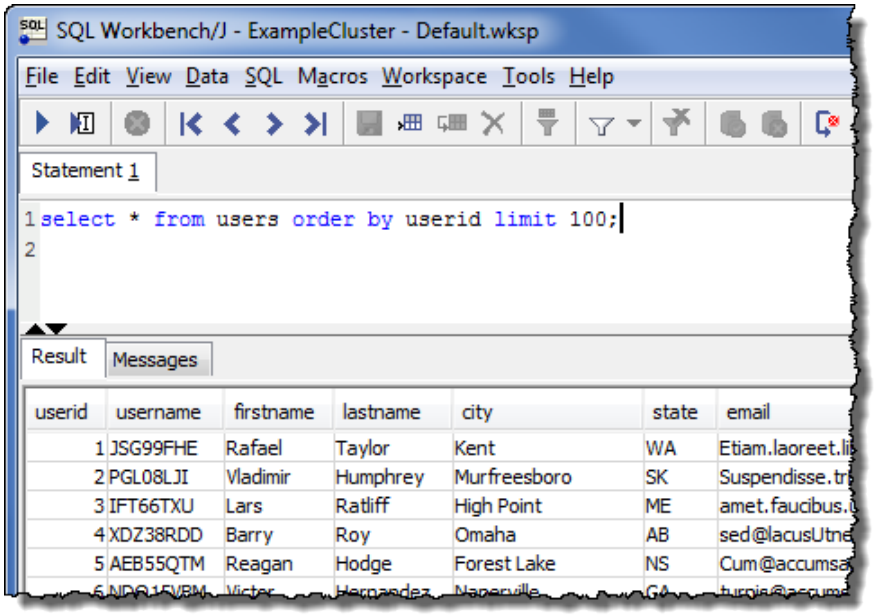
Test the SQL Workbench/J Connection

After you configure your JDBC or ODBC connection, you can test the connection by running an example query.

1. If you loaded the sample tables and data from the Amazon Redshift Getting Started, you can test your connection by typing the following query into the **Statement** window:

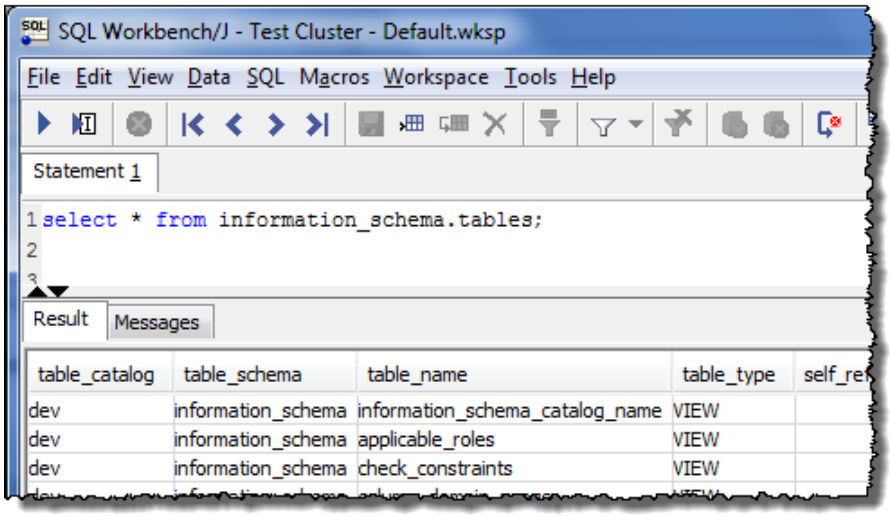
   ```
   select * from users order by userid limit 100;
   ```

   If your connection is successful, a listing of records appears in the **Results** tab.


SQL Workbench/J - ExampleCluster - Default.wksp
File Edit View Data SQL Macros Workspace Tools Help
Statement 1
1 select * from users order by userid limit 100;
2
Result
Messages
userid username firstname lastname city state email
1 JSG99FHE Rafael Taylor Kent WA Etiam.laoreet.li
2 PGL08LJI Vladimir Humphrey Murfreesboro SK Suspendisse.tr
3 IFT66TXU Lars Ratliff High Point ME amet.faucibus.
4 XDZ38RDD Barry Roy Omaha AB sed@lacusUtne
5 AEB55QTM Reagan Hodge Forest Lake NS Cum@accumsa

2. Alternatively, you can use the following query to test your connection.

   ```
   select * from information_schema.tables;
   ```



Connect to Your Cluster by Using the psql Tool

After you create an Amazon Redshift cluster, you can use psql, a terminal-based front end from PostgreSQL, to query the data in your cluster. You can type the queries interactively or read them from a file. To connect from psql, you must specify the cluster endpoint, database, and port.

> **Note**
> Amazon Redshift does not provide the psql tool; it is installed with PostgreSQL. For information about using psql, go to http://www.postgresql.org/docs/8.4/static/app-psql.html. For information about installing the PostgreSQL client tools, select your operating system from the PostrgreSQL binary downloads page at http://www.postgresql.org/download/.

Topics

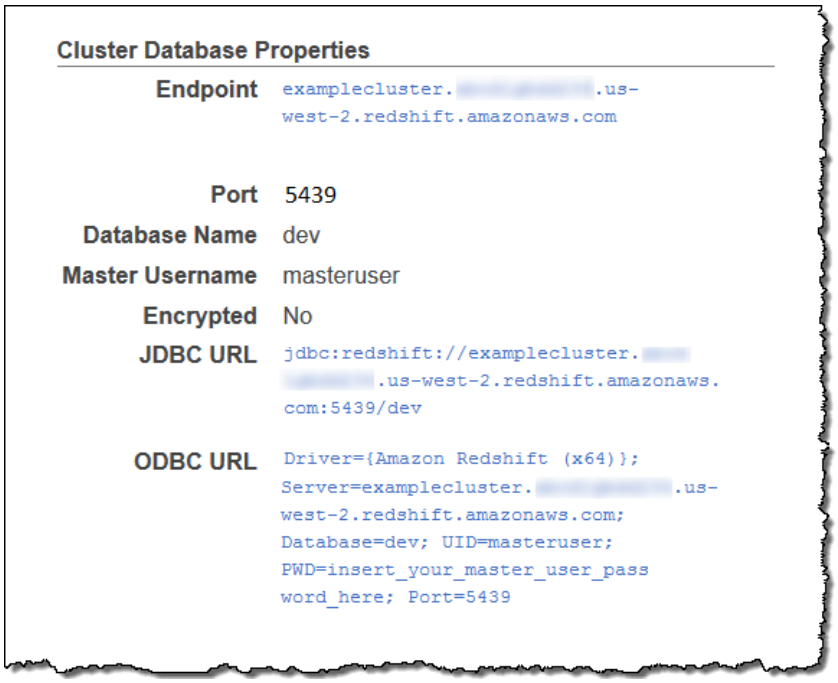
- Connect by Using the psql Defaults (p. 166)
- Connect by Using a Certificate (p. 167)

Connect by Using the psql Defaults

By default, psql does not validate the Amazon Redshift service; it makes an encrypted connection by using Secure Sockets Layer (SSL).

To connect by using psql defaults

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. In the left navigation pane, click **Clusters**. Click your cluster to open it. Under **Cluster Database Properties**, record the values of **Endpoint**, **Port**, and **Database Name**.



3. At a command prompt, specify the connection information by using either command line parameters or a connection information string. To use parameters:

```
psql -h <endpoint> -U <userid> -d <databasename> -p <port>
```

Where:

- *<endpoint>* is the **Endpoint** you recorded in the previous step.
- *<userid>* is a user ID with permissions to connect to the cluster.
- *<databasename>* is the **Database Name** you recorded in the previous step.
- *<port>* is the **Port** you recorded in the previous step.

For example:

```
psql -h examplecluster.<XXXXXXXXXXXX>.us-west-2.redshift.amazonaws.com -U
masteruser -d dev -p 5439
```

4. At the psql password prompt, enter the password for the *<userid>* user.

You are connected to the cluster, and you can interactively enter commands.

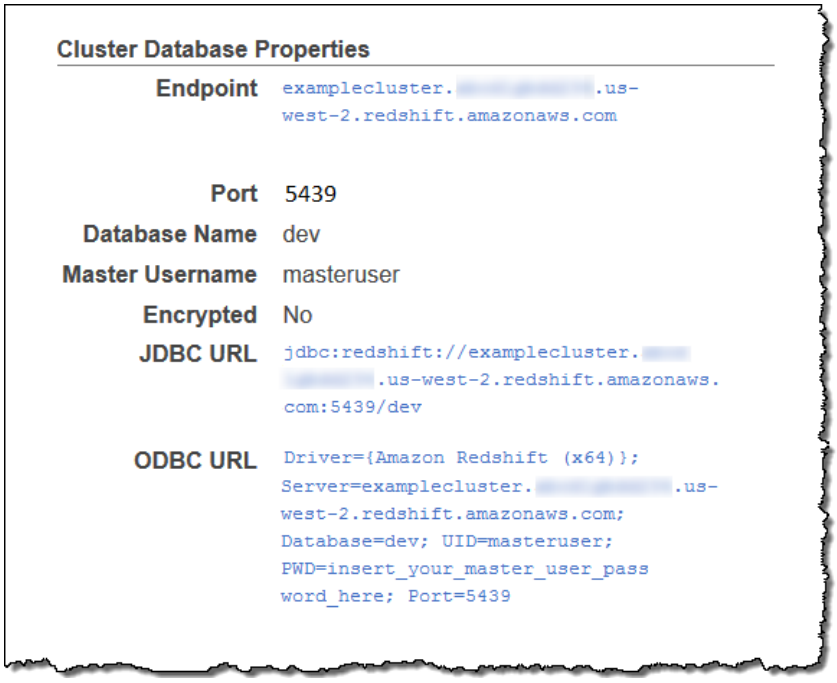
Connect by Using a Certificate

To control whether psql authenticates the service using a certificate, you must use a connection information string to specify connection information, and specify the *sslmode* keyword. By default, psql operates with *sslmode=prefer*. To specify that psql opens an encrypted connection and uses an Amazon Redshift certificate to verify the service, download an Amazon Redshift certificate to your computer, specify *sslmode=verify-ca*, and specify *sslrootcert* with the location of the certificate. For more information about sslmode, see Configure Security Options for Connections (p. 158).

For more information about connection information string parameters, see http://www.postgresql.org/docs/8.4/static/libpq-connect.html.

To connect by using a certificate

1. Save the public key from https://s3.amazonaws.com/redshift-downloads/redshift-ssl-ca-cert.pem as a .pem file to your computer. If you do a **File\Save as** using Internet Explorer, specify the file type as **Text file (*.txt)** and delete the .txt extension. For example, save it as the file `C:\MyDownloads\redshift-ssl-ca-cert.pem`.
2. In the Amazon Redshift console, select the cluster to display the **Cluster Database Properties**. Record the values displayed in the **Endpoint**, **Port**, and **Database Name** fields.



3. At a command prompt, specify the connection information using a connection information string:

```
psql "host=<endpoint> user=<userid> dbname=<databasename> port=<port> sslmode=verify-ca sslrootcert=<certificate>"
```

Where:

- *<endpoint>* is the **Endpoint** you recorded in the previous step.
- *<userid>* is a user ID with permissions to connect to the cluster.
- *<databasename>* is the **Database Name** you recorded in the previous step.
- *<port>* is the **Port** you recorded in the previous step.
- *<certificate>* is the full path to the certificate file. On Windows systems, the certificate path must be specified using Linux-style / separators instead of the Windows \ separator.

For example:

```
psql "host=examplecluster.<XXXXXXXXXXXX>.us-west-2.redshift.amazonaws.com user=masteruser dbname=dev port=5439 sslmode=verify-ca sslrootcert=C:/MyDownloads/redshift-ssl-ca-cert.pem"
```

4. At the psql password prompt, enter the password for the *<userid>* user.

You are connected to the cluster, and you can interactively enter commands.

Connect to Your Cluster Programmatically

Topics

- Connecting to a Cluster by Using Java (p. 169)
- Connecting to a Cluster by Using .NET (p. 171)

This section explains how to connect to your cluster programmatically. If you are using an application like SQL Workbench/J that manages your client connections for you, then you can skip this section.

Connecting to a Cluster by Using Java

When you use Java to programmatically connect to your cluster, you can do so with or without server authentication. If you plan to use server authentication, follow the instructions in Configure Security Options for Connections (p. ?) to put the Amazon Redshift server certificate into a keystore. You can refer to the keystore by specifying a property when you run your code as follows:

```
-Djavax.net.ssl.trustStore=<path to keystore>
-Djavax.net.ssl.trustStorePassword=<keystore password>
```

Example : Connect to a Cluster by Using Java

The following example connects to a cluster and runs a sample query that returns system tables. It is not necessary to have data in your database to use this example.

If you are using a server certificate to authenticate your cluster, you can restore the line that uses the keystore, which is commented out:

```
props.setProperty("ssl", "true");
```

For more information about the server certificate, see Configure Security Options for Connections (p. 158).

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130).

```
package connection;

import java.sql.*;
import java.util.Properties;

public class Docs {
    //Redshift driver: "jdbc:redshift://x.y.us-west-2.redshift.amazon
aws.com:5439/dev";
 //or "jdbc:postgresql://x.y.us-west-2.redshift.amazonaws.com:5439/dev";
    static final String dbURL = "***jdbc cluster connection string ****";
    static final String MasterUsername = "***master user name***";
    static final String MasterUserPassword = "***master user password***";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
           //Dynamically load driver at runtime.
           //Redshift JDBC 4.1 driver: com.amazon.redshift.jdbc41.Driver
           //Redshift JDBC 4 driver: com.amazon.redshift.jdbc4.Driver
           Class.forName("com.amazon.redshift.jdbc4.Driver");

           //Open a connection and define properties.
           System.out.println("Connecting to database...");
           Properties props = new Properties();

           //Uncomment the following line if using a keystore.
           //props.setProperty("ssl", "true");
           props.setProperty("user", MasterUsername);
           props.setProperty("password", MasterUserPassword);
           conn = DriverManager.getConnection(dbURL, props);

           //Try a simple query.
           System.out.println("Listing system tables...");
           stmt = conn.createStatement();
           String sql;
           sql = "select * from information_schema.tables;";
           ResultSet rs = stmt.executeQuery(sql);

           //Get the data from the result set.
           while(rs.next()){
```

```
            //Retrieve two columns.
            String catalog = rs.getString("table_catalog");
            String name = rs.getString("table_name");

            //Display values.
            System.out.print("Catalog: " + catalog);
            System.out.println(", Name: " + name);
         }
         rs.close();
         stmt.close();
         conn.close();
      }catch(Exception ex){
         //For convenience, handle all errors here.
         ex.printStackTrace();
      }finally{
         //Finally block to close resources.
         try{
            if(stmt!=null)
               stmt.close();
         }catch(Exception ex){
         }// nothing we can do
         try{
            if(conn!=null)
               conn.close();
         }catch(Exception ex){
            ex.printStackTrace();
         }
      }
      System.out.println("Finished connectivity test.");
   }
}
```

Connecting to a Cluster by Using .NET

When you use .NET (C#) to programmatically connect to your cluster, you can do so with or without server authentication. If you plan to use server authentication, follow the instructions in Configure Security Options for Connections (p. ?) to download the Amazon Redshift server certificate, and then put the certificate in the correct form for your .NET code.

Example Connect to a Cluster by Using .NET

The following example connects to a cluster and runs a sample query that returns system tables. It does not show server authentication. It is not necessary to have data in your database to use this example. This example uses the System.Data.Odbc Namespace, a .NET Framework Data Provider for ODBC.

```
using System;
using System.Data;
using System.Data.Odbc;

namespace redshift.amazon.com.docsamples
{
    class ConnectToClusterExample
    {
        public static void Main(string[] args)
        {

            DataSet ds = new DataSet();
            DataTable dt = new DataTable();

          // Server, e.g. "examplecluster.xyz.us-west-2.redshift.amazonaws.com"

            string server = "***provide server name part of connection
string****";

            // Port, e.g. "5439"
            string port = "***provide port***";

            // MasterUserName, e.g. "masteruser".
            string masterUsername = "***provide master user name***";

            // MasterUserPassword, e.g. "mypassword".
            string masterUserPassword = "***provide master user password***";

            // DBName, e.g. "dev"
            string DBName = "***provide name of database***";

            string query = "select * from information_schema.tables;";

            try
            {
                // Create the ODBC connection string.
                //Redshift ODBC Driver - 64 bits
                /*
                string connString = "Driver={Amazon Redshift (x64)};" +
                    String.Format("Server={0};Database={1};" +
                    "UID={2};PWD={3};Port={4};SSL=true;Sslmode=Require",
                    server, DBName, masterUsername,
                    masterUserPassword, port);
                */

                //Redshift ODBC Driver - 32 bits
                string connString = "Driver={Amazon Redshift (x86)};" +
                    String.Format("Server={0};Database={1};" +
                    "UID={2};PWD={3};Port={4};SSL=true;Sslmode=Require",
                    server, DBName, masterUsername,
                    masterUserPassword, port);
```

```
                // Make a connection using the psqlODBC provider.
                OdbcConnection conn = new OdbcConnection(connString);
                conn.Open();

                // Try a simple query.
                string sql = query;
                OdbcDataAdapter da = new OdbcDataAdapter(sql, conn);
                da.Fill(ds);
                dt = ds.Tables[0];
                foreach (DataRow row in dt.Rows)
                {
                    Console.WriteLine(row["table_catalog"] + ", " +
row["table_name"]);
                }

                conn.Close();
                Console.ReadKey();
            }
            catch (Exception ex)
            {
                Console.Error.WriteLine(ex.Message);
                Console.ReadKey();
            }

        }
    }
}
```

Troubleshooting Connection Issues in Amazon Redshift

If you have issues with connecting to your cluster from a SQL client tool, there are several things that you can check to narrow down the problem. If you are using SSL or server certificates, first remove this complexity while you troubleshoot the connection issue. Then add this back when you have found a solution. For more information, see Configure Security Options for Connections (p. 158).

The following section has some example error messages and possible solutions for connection issues. Because different SQL client tools provide different error messages, this is not a complete list, but should be a good starting point for troubleshooting issues.

Topics

- Connecting from Outside of Amazon EC2—Firewall Timeout Issue (p. 173)
- The Connection Is Refused or Fails (p. 175)
- The Client and Driver Are Incompatible (p. 176)
- Queries Appear to Hang in the Client and Do Not Reach the Cluster (p. 176)
- Queries Appear to Hang in the Client and the Cluster (p. 177)

Connecting from Outside of Amazon EC2—Firewall Timeout Issue

Example issue:

Your client connection to the database appears to hang or timeout when running long queries, such as a COPY command. In this case, you might observe that the Amazon Redshift console displays that the query has completed, but the client tool itself still appears to be running the query. The results of the query might be missing or incomplete depending on when the connection stopped.

Possible solutions:

This happens when you connect to Amazon Redshift from a computer other than an Amazon EC2 instance, and idle connections are terminated by an intermediate network component, such as a firewall, after a period of inactivity. This behavior is typical when you log in from a Virtual Private Network (VPN) or your local network.

To avoid these timeouts, we recommend the following changes:

- Increase client system values that deal with TCP/IP timeouts. You should make these changes on the computer you are using to connect to your cluster. The timeout period should be adjusted for your client and network. See To change TCP/IP timeout settings (p. 174).
- Optionally, set keep-alive behavior at the DSN level. See To change DSN timeout settings (p. 175).

To change TCP/IP timeout settings

- Configure the timeout settings according to the operating system that you use to connect to your cluster.

  a. Linux — If your client is running on Linux, run the following command as the root user.

  ```
  /sbin/sysctl -w net.ipv4.tcp_keepalive_time=200
  net.ipv4.tcp_keepalive_intvl=200 net.ipv4.tcp_keepalive_probes=5
  ```

  b. Windows — If your client runs on Windows, edit the values for the following registry settings under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\:

  - KeepAliveTime: 30000
  - KeepAliveInterval: 1000
  - TcpMaxDataRetransmissions: 10

  These settings use the DWORD data type. If they do not exist under the registry path, you can create the settings and specify these recommended values. For more information about editing the Windows registry, refer to Windows documentation.

  After you set these values, restart your computer for the changes to take effect.

  c. Mac — If your client is running on a Mac, create or modify the /etc/sysctl.conf file with the following values:

  ```
  net.inet.tcp.keepidle=20000
  net.inet.tcp.keepintvl=20000
  net.inet.tcp.keepinit=20000
  net.inet.tcp.always_keepalive=1
  ```

  Restart your computer, and then run the following commands to verify the values are set:

```
sysctl net.inet.tcp.keepidle
sysctl net.inet.tcp.keepintvl
sysctl net.inet.tcp.keepinit
sysctl net.inet.tcp.always_keepalive
```

To change DSN timeout settings

You can set keep-alive behavior at the DSN level if you choose. You do this by adding or modifying the following parameters in the odbc.ini file:

KeepAlivesCount
The number of TCP keep-alive packets that can be lost before the connection is considered broken.

KeepAlivesIdle
The number of seconds of inactivity before the driver sends a TCP keep-alive packet.

KeepAlivesInterval
The number of seconds between each TCP keep-alive retransmission.

On Windows, you modify these parameters in the registry by adding or changing keys in HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*your_DSN*. On Linux and Mac OS, you add or modify these parameters in the target DSN entry directly in the odbc.ini file. For more information on modifying the odbc.ini file on Linux and Mac OS computers, see Configure the ODBC Driver on Linux and Mac OS X Operating Systems (p. 151).

If these parameters don't exist, or if they have a value of 0, the system will use the keep-alive parameters specified for TCP/IP to determine DSN keep-alive behavior. On Windows, the TCP/IP parameters can be found in the registry in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\. On Linux and Mac OS, the TCP/IP parameters can be found in the sysctl.conf file.

The Connection Is Refused or Fails

Example errors:

- "Failed to establish a connection to *<endpoint>*."
- "Could not connect to server: Connection timed out. Is the server running on host *'<endpoint>'* and accepting TCP/IP connections on port *'<port>'*?"
- "Connection refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections."

Possible solutions:

Generally, when you receive an error message indicating that there is a failure to establish a connection, it is an issue with permission to access the cluster.

If you attempt to connect to the cluster from a client tool outside of the network that the cluster is in, you must add an ingress rule to the cluster security group for the CIDR/IP that you are connecting from:

- If you created your Amazon Redshift cluster in a VPC, you need to add your client CIDR/IP address to the VPC security group in Amazon VPC. For more information about configuring VPC security groups for your cluster, see Managing Clusters in an Amazon Virtual Private Cloud (VPC) (p. 32).
- If you created your Amazon Redshift cluster outside a VPC, you need to add your client CIDR/IP address to the cluster security group in Amazon Redshift. For more information about configuring cluster security groups, see Amazon Redshift Cluster Security Groups (p. 43).

If you attempt to connect to the cluster from a client tool in an Amazon EC2 instance, you must add an ingress rule to the cluster security group for the Amazon EC2 security group that is associated with the Amazon EC2 instance. For more information about configuring cluster security groups, see Amazon Redshift Cluster Security Groups (p. 43).

Additionally, if you have a layer between your client and server, such as a firewall, make sure that the firewall accepts inbound connections over the port that you configured for your cluster.

The Client and Driver Are Incompatible

Example error:

"The specified DSN contains an architecture mismatch between the Driver and Application."

Possible solution:

When you get attempt to connect and get an error about an architecture mismatch, this means that the client tool and the driver are not compatible because their system architecture does not match. For example, this can happen if you have a 32-bit client tool but have installed the 64-bit version of the driver. Sometimes 64-bit client tools can use 32-bit drivers, but you cannot use 32-bit applications with 64-bit drivers. Make sure that the driver and client tool are using the same version of the system architecture.

Queries Appear to Hang in the Client and Do Not Reach the Cluster

Example issue:

Users experience issues with their queries completing. The queries appear to be running and hang in the user's SQL client tool, but they aren't logged or found running in the system tables and Amazon Redshift console.

Possible solution:

Set the size of the maximum transmission unit (MTU) to 1500 on the network interface (NIC) of the Amazon EC2 instances from which you connect to your Amazon Redshift cluster. The MTU size determines the maximum size, in bytes, of a packet that can be transferred in one Ethernet frame over your network connection. If your MTU size is too large for the connection, you might experience incomplete query results, your query might hang, or the connection might be dropped altogether. Some Amazon EC2 instance types provide 1500 MTU (Ethernet v2 frames), and other instance types provide 9001 MTU (TCP/IP jumbo frames). We recommend always disabling TCP/IP jumbo frames to ensure that connections consistently use the same packet size.

Set MTU on a Microsoft Windows Operating System

If your client runs in a Microsoft Windows operating system, you can review and set the MTU value for the Ethernet adapter by using the `netsh` command. Run the following command to determine the current MTU value:

```
netsh interface ipv4 show subinterfaces
```

Review the `MTU` value for the `Ethernet` adapter in the output. If the value is not `1500`, run the following command to set it:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
```

After you set this value, restart your computer for the changes to take effect.

Set MTU on a Linux Operating System

If your client runs in a Linux operating system, you can review and set the MTU value by using the `ip` command. Run the following command to determine the current MTU value:

```
$ ip link show eth0
```

Review the value following `mtu` in the output. If the value is not `1500`, run the following command to set it:

```
$ ip link set dev eth0 mtu 1500
```

Set MTU on Mac OS X

To set the MTU on a Mac operating system, follow the instructions in Mac OS X 10.4 or later: How to change the MTU for troubleshooting purposes.

For a similar issue with a different solution, see Queries Appear to Hang in the Client and the Cluster (p. 177).

Queries Appear to Hang in the Client and the Cluster

Example issue:

Users experience issues with their queries completing. The queries run on the user's SQL client tool and reach the database, but they don't complete and sometimes can't be terminated.

Possible solution:

Make sure that users configure their SQL client tools to automatically commit SQL commands, and that they use a transaction block (defined by BEGIN and END statements) to enclose a set of commands. This configuration ensures that each block commits as one transaction that you can roll back if necessary.

The example issue can occur when a user submits a transaction without committing the block as one transaction, and an error occurs during processing. If the client tool attempts to roll back to a previous point inside the transaction (called a *savepoint*), the attempt fails because Amazon Redshift does not support rolling back to a savepoint. As a result, the transaction cannot proceed. If the PG_TERMINATE_BACKEND function fails to terminate the connection from which the transaction was submitted, the locks on the relevant user tables cannot be released. In some cases, this failure can cause the cluster to become unavailable and require a reboot.

When this issue happens, users see error such as `ERROR: SQL command "ROLLBACK TO` *`savepoint_name`*`;" not supported` and `ERROR: current transaction is aborted, commands ignored until end of transaction block.`

Many client tools, such as SQL Workbench/J, enable the use of savepoints by default. Because savepoints are incompatible with Amazon Redshift, users must configure their client tools to bypass this behavior. You can run the following query to determine which users are using SQL client tools that create savepoints.

```
SELECT ut.userid,
trim(u.usename),
COUNT(text) as count
FROM stl_utilitytext ut
JOIN pg_user u ON ut.userid = u.usesysid
WHERE ut.text LIKE 'SAVEPOINT%'
GROUP BY ut.userid, u.usename
ORDER BY count DESC;
```

When you know which users have SQL client tools that create savepoints, work with these users to configure their client tools:

- In SQL Workbench/J and some other tools, users can prevent attempts to roll back to a savepoint by making sure that transactions are automatically committed. To configure this functionality in SQL Workbench/J, enable the **Autocommit** option in the connection profile. For more information about configuring this option, see Connect to Your Cluster by Using SQL Workbench/J (p. 162).
- For client tools other than SQL Workbench/J, refer to the tool's documentation to either enable automatic commit functionality or to disable savepoints. Either solution prevents rolling back to savepoints.

For a similar issue with a different solution, see Queries Appear to Hang in the Client and Do Not Reach the Cluster (p. 176).

Monitoring Amazon Redshift Cluster Performance

Topics

- Overview (p. 179)
- Summary of Amazon Redshift Performance Data (p. 180)
- Working with Performance Data in the Amazon Redshift Console (p. 183)

Overview

Amazon Redshift provides performance metrics and data so that you can track the health and performance of your clusters and databases. In this section, we discuss the types of data you can work with in Amazon Redshift and specifically, in the Amazon Redshift console. The performance data that you can use in Amazon Redshift console falls into two categories:

- **Amazon CloudWatch Metrics —** Amazon CloudWatch metrics help you monitor physical aspects of your cluster, such as CPU utilization, latency, and throughput. Metric data is displayed directly in the Amazon Redshift console. You can also view it in the Amazon CloudWatch console, or you can consume it in any other way you work with metrics such as with the Amazon CloudWatch Command Line Interface (CLI) or one of the AWS Software Development Kits (SDKs).
- **Query/Load Performance Data —** Performance data helps you monitor database activity and performance. This data is aggregated in the Amazon Redshift console to help you easily correlate what you see in Amazon CloudWatch metrics with specific database query and load events. You can also create your own custom performance queries and run them directly on the database. Query and load performance data is displayed only in the Amazon Redshift console. It is not published as Amazon CloudWatch metrics.

Performance data is integrated into the Amazon Redshift console, yielding a richer experience in the following ways:

- Performance data associated with a cluster is displayed contextually when you view a cluster, where you might need it to make decisions about the cluster such as resizing.
- Some performance metrics are displayed in more appropriately scaled units in the Amazon Redshift console as compared to Amazon CloudWatch. For example, `WriteThroughput`, is displayed in GB/s

(as compared to Bytes/s in Amazon CloudWatch), which is a more relevant unit for the typical storage space of a node.
- Performance data for the nodes of a cluster can easily be displayed together on the same graph so that you can easily monitor the performance of all nodes of a cluster; however, you can also view performance data per node.

Amazon Redshift provides performance data (both Amazon CloudWatch metrics and query and load data) at no additional charge. Performance data is recorded every minute. You can access historical values of performance data in the Amazon Redshift console. For detailed information about using Amazon CloudWatch to access the Amazon Redshift performance data that is exposed as Amazon CloudWatch metrics, go to the Introduction to Amazon CloudWatch in the *Amazon CloudWatch Developer Guide*.

Summary of Amazon Redshift Performance Data

Amazon Redshift CloudWatch Metrics

Amazon Redshift CloudWatch metrics enable you to get information about your cluster's health and performance, and to drill down and see that information at the node level. When working with these metrics, you should keep in mind that each metric has one or more dimensions associated with it that tell you what the metric is applicable to, that is the scope of the metric. Amazon Redshift has the following two dimensions:

- Metrics that have a `NodeID` dimension are metrics that provide performance data for nodes of a cluster. This includes leader and compute nodes. Examples of these metrics include `CPUUtilization`, `ReadIOPS`, `WriteIOPS`.
- Metrics that have just a `ClusterIdentifier` dimension are metrics that provide performance data for clusters. Examples of these metrics include `HealthStatus` and `MaintenanceMode`.

> **Note**
> In some metric cases, a cluster-specific metric represents an aggregation of node behavior and care must be taken in the interpretation of the metric value because the leader node's behavior is aggregated with the compute node.

For more information about Amazon CloudWatch metrics and dimensions, go to Amazon CloudWatch Concepts in the *Amazon CloudWatch Developer Guide*.

The following table describes all the metrics available for you to use.

Amazon Redshift Metrics

The following metrics are available from Amazon Redshift.

Metric	Description
`CPUUtilization`	The percentage of CPU utilization. For clusters, this metric represents an aggregation of all nodes (leader and compute) CPU utilization values. Units: Percent Dimensions: `NodeID`, `ClusterIdentifier`

Metric	Description
DatabaseConnections	The number of database connections to a cluster. Units: Count Dimensions: ClusterIdentifier
HealthStatus	Indicates the health of the cluster. Every minute the cluster connects to its database and performs a simple query. If it is able to perform this operation successfully, the cluster is considered healthy. Otherwise, the cluster is unhealthy. An unhealthy status can occur when the cluster database is under extremely heavy load or if there is a configuration problem with a database on the cluster. The exception to this is when the cluster is undergoing maintenance. Even though your cluster might be unavailable due to maintenance tasks, the cluster remains in HEALTHY state. For more information, see Maintenance Windows in the *Amazon Redshift Cluster Management Guide*. **Note** In Amazon CloudWatch this metric is reported as 1 or 0 whereas in the Amazon Redshift console, this metric is displayed with the words HEALTHY or UNHEALTHY for convenience. When this metric is displayed in the Amazon Redshift console, sampling averages are ignored and only HEALTHY or UNHEALTHY are displayed. In Amazon CloudWatch, values different than 1 and 0 may occur because of sampling issue. Any value below 1 for HealthStatus is reported as 0 (UNHEALTHY). Units: 1/0 (HEALTHY/UNHEALTHY in the Amazon Redshift console) Dimensions: ClusterIdentifier
MaintenanceMode	Indicates whether the cluster is in maintenance mode. **Note** In Amazon CloudWatch this metric is reported as 1 or 0 whereas in the Amazon Redshift console, this metric is displayed with the words ON or OFF for convenience. When this metric is displayed in the Amazon Redshift console, sampling averages are ignored and only ON or OFF are displayed. In Amazon CloudWatch, values different than 1 and 0 may occur because of sampling issues. Any value greater than 0 for MaintenanceMode is reported as 1 (ON). Units: 1/0 (ON/OFF in the Amazon Redshift console). Dimensions: ClusterIdentifier
NetworkReceiveThroughput	The rate at which the node or cluster receives data. Units: Bytes/seconds (MB/s in the Amazon Redshift console) Dimensions: NodeID, ClusterIdentifier
NetworkTransmitThroughput	The rate at which the node or cluster writes data. Units: Bytes/second (MB/s in the Amazon Redshift console) Dimensions: NodeID, ClusterIdentifier

Metric	Description
PercentageDiskSpaceUsed	The percent of disk space used. Units: Percent Dimensions: NodeID, ClusterIdentifier
ReadIOPS	The average number of disk read operations per second. Units: Count/second Dimensions: NodeID
ReadLatency	The average amount of time taken for disk read I/O operations. Units: Seconds Dimensions: NodeID
ReadThroughput	The average number of bytes read from disk per second. Units: Bytes (GB/s in the Amazon Redshift console) Dimensions: NodeID
WriteIOPS	The average number of write operations per second. Units: Count/seconds Dimensions: NodeID
WriteLatency	The average amount of time taken for disk write I/O operations. Units: Seconds Dimensions: NodeID
WriteThroughput	The average number of bytes written to disk per second. Units: Bytes (GB/s in the Amazon Redshift console) Dimensions: NodeID

Amazon Redshift Query/Load Performance Data

In addition to the Amazon CloudWatch metrics, Amazon Redshift provides query and load performance data. Query and load performance data can be used to help you understand the relation between database performance and cluster metrics. For example, if you notice that a cluster's CPU spiked, you can find the spike on the cluster CPU graph and see the queries that were running at that time. Conversely, if you are reviewing a specific query, metric data (like CPU) is displayed in context so that you can understand the query's impact on cluster metrics.

Query and load performance data are not published as Amazon CloudWatch metrics and can only be viewed in the Amazon Redshift console. Query and load performance data are generated from querying with your database's system tables (see System Tables Reference in the *Amazon Redshift Developer Guide*). You can also generate your own custom database performance queries, but we recommend starting with the query and load performance data presented in the console. For more information about measuring and monitoring your database performance yourself, see Managing Performance in the *Amazon Redshift Developer Guide*

The following table describes different aspects of query and load data you can access in the Amazon Redshift console.

Query/Load Data	Description
Query summary	A list of queries in a specified time period. The list can be sorted on values such as query ID, query run time, and status. Access this data in the **Queries** tab of the cluster detail page.
Query Detail	Provides details on a particular query including: • Query properties such as the query ID, type, cluster the query was run on, and run time. • Details such as the status of the query and the number of errors. • The SQL statement that was run. • An explain plan if available. • Cluster performance data during the query execution (see Amazon Redshift CloudWatch Metrics (p. 180)).
Load Summary	Lists all the loads in a specified time period. The list can be sorted on values such as query ID, query run time, and status. Access this data in the **Loads** tab of the cluster detail page. Access this data in the **Queries** tab of the cluster detail page.
Load Detail	Provides details on a particular load operation including: • Load properties such as the query ID, type, cluster the query was run on, and run time. • Details such as the status of the load and the number of errors. • The SQL statement that was run. • A list of loaded files. • Cluster performance data during the load operation (see Amazon Redshift CloudWatch Metrics (p. 180)).

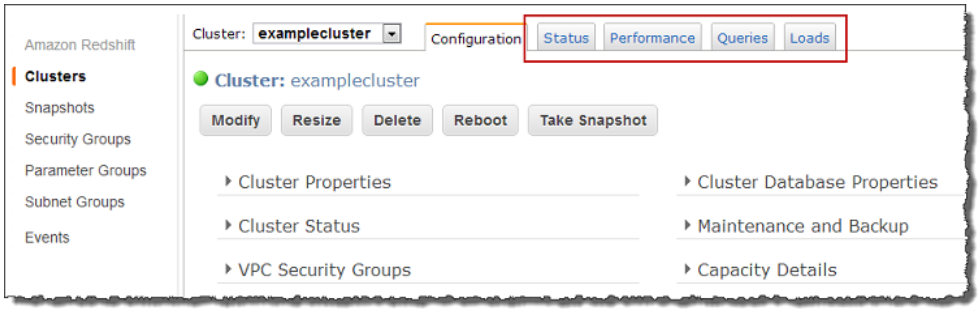
Working with Performance Data in the Amazon Redshift Console

This section explains how to view performance data in the Amazon Redshift console which includes information about cluster and query performance. Additionally, you can create alarms on cluster metrics directly from the Amazon Redshift console.

When you view performance data in the Amazon Redshift console, you view it by cluster. The performance data graphs for a cluster are designed to give you access to data to answer your most common performance questions. For some performance data (see Amazon Redshift CloudWatch Metrics (p. 180)), you can also use Amazon CloudWatch to further customize your metrics graphs, for example, choose longer times or combine metrics across clusters. For more information about working with the Amazon CloudWatch console, see Working with Performance Metrics in the Amazon CloudWatch Console (p. 197).

To start working with performance data find your cluster in the *cluster performance dashboard*. The dashboard is a list of clusters that shows at a glance the status of the cluster (e.g. **available**), the **DB**

Health of the cluster (e.g. **healthy**), whether the cluster is undergoing maintenance, and count of recent events. From the dashboard, select a cluster to work with and go to the details of the cluster. From this page you can access the **Events+Alarms**, **Performance**, **Queries**, and **Loads** tabs which contain the performance data. These tabs are shown in the following example.



Viewing Cluster Performance Data

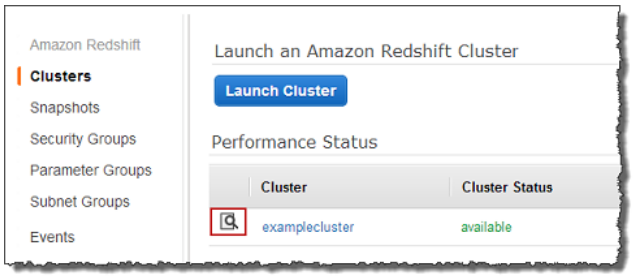
Cluster metrics in Amazon Redshift enable the following common performance use cases:

- Determine if cluster metrics are abnormal over a specified time range and, if so, identify the queries responsible for the performance hit.
- Check if historical or current queries are impacting cluster performance. If you identify a problematic query, you can be view details about it including the cluster performance during the query's execution, information which may assist you in diagnosing why the query was slow, and what can be done to improve its performance.

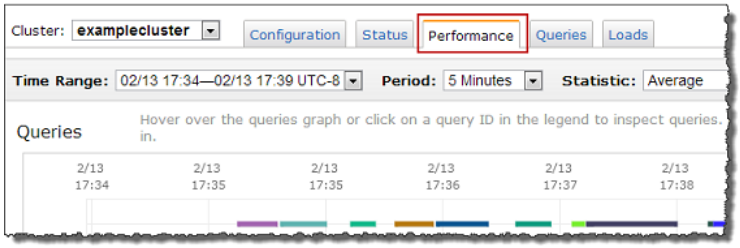
The default cluster view shows all nodes graphed together, an `Average` statistic, and data for the last hour. You can change this view as needed. Some metrics, such as `HealthStatus`, are only applicable for the leader node while others, such as `WriteOps`, are only applicable for compute nodes. Switching the node display mode will reset all filters.

To view cluster performance data

1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation, click **Clusters**.
3. In the **Cluster** list, click the magnifying glass icon beside the cluster for which you want to view performance data.



4. Select the **Performance** tab.

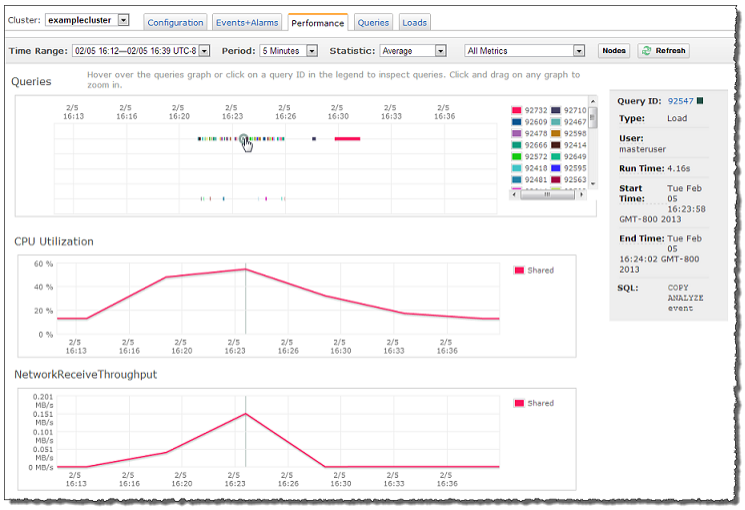


By default, the performance view displays cluster performance over the past hour. If you need to fine tune the view you have *filters* that you can use as described in the following table.

To...	Use this filter...
Change the time range for which data is displayed	Select a time range from the **Time Range** drop down. By default, the last hour is shown.
Change the period for which data is displayed	Select a period from the **Period** drop down. By default, a 5 minute period is shown. Use a period smaller than 5 minutes if you need more detail when investigating a metric (drilling in) and displaying metrics over a small time period, for example 10 minutes. Similarly, use a period greater than 5 minutes when viewing metrics over a large period of time, for example, days.
Change the statistic that is displayed for metrics	Select a statistic from the **Statistic** drop down. By default, the `Average` statistic is used.
Change what metrics are shown, all or a specific metric	Select a metrics from the **Metrics** drop down. By default, all metrics are shown.
Change whether node metrics are displayed separately or together on the same graph	Click the **Nodes** button. By default, node data for a given metric is shown on a combined graph. If you choose to display node data on separate graphs, you can additionally show or hide individual nodes.

Cluster Metrics: Examples

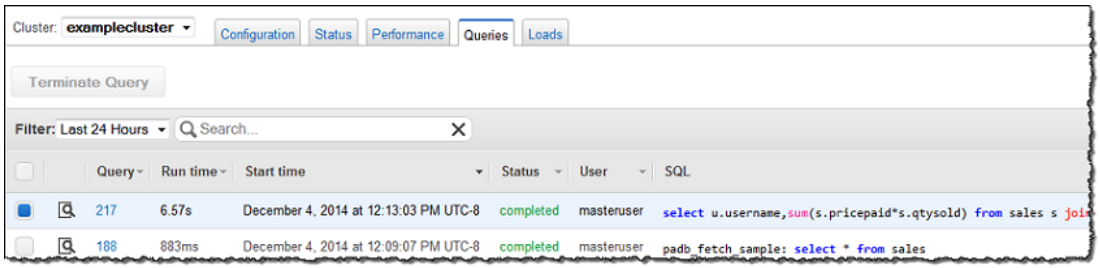
The following example shows `CPUUtilization` and `NetworkReceiveThroughput` metrics for a single node cluster. In this case the graphs for cluster metrics show one line marked as **Shared** since the leader and compute node are combined. The example shows that multiple queries were run in the time period shown. On the **Queries** graph the cursor is positioned over the query running at the peak values of the two metrics and the **Query ID** is displayed on the right. You could then click the Query ID to find out more about the query running.



The following example shows the `NetworkReceiveThroughput` for a cluster with two nodes. It shows a line for the leader and two compute nodes. Note that the leader node metrics is flat and is not of interest since data is only loaded on the compute nodes. The example shows that one long query ran in the time period shown. On the **Queries** graph the cursor is positioned over the long running query and the **Query ID** is displayed on the right. You could then click the **Query ID** to find out more about the query running. The `NetworkReceiveThroughput` value is displayed during the query execution.

Viewing Query Performance Data

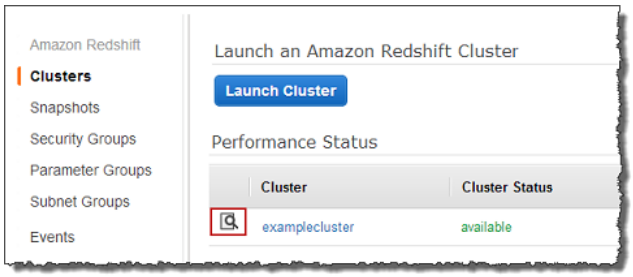
The Amazon Redshift console provides information about performance of queries that run in the database. You can use this information to identify and troubleshoot queries that take a long time to process and that create bottlenecks preventing other queries from processing efficiently. You can use the **Queries** tab on the cluster details page to view this information. The **Queries** tab shows a table that lists queries that are currently running or have run recently in the cluster.



Use the button bar, shown following, to refresh the data in the table, to configure the columns that appear in the table, or to open the Amazon Redshift documentation.

To view query performance data

1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation, click **Clusters**.
3. In the **Cluster** list, click the magnifying glass icon beside the cluster for which you want to view performance data.



4. Select the **Queries** tab.

   By default, the **Queries** tab displays query performance over the past 24 hours. To change the data displayed, use the **Filter** list to select the time period for which you want to view queries, or type a keyword in the **Search** box to search for queries that match your search criteria.

Terminating a Running Query

You can also use the **Queries** page to terminate a query that is currently in progress.

> **Note**
> The ability to terminate queries and loads in the Amazon Redshift console requires specific permission. If you select the **Amazon Redshift Read Only** AWS managed policy or create a custom policy in IAM, and you want users to be able to terminate queries and loads, make sure to add the `redshift:CancelQuerySession` action to the policy. Users who have the **Amazon Redshift Full Access** policy already have the necessary permission to terminate queries and loads. For more information about actions in IAM policies for Amazon Redshift, see IAM Policy Actions for Amazon Redshift (p. 118).

To terminate a running query

1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Clusters**.

3. In the **Cluster** list, click the cluster you want to open.
4. Click the **Queries** tab.
5. Do one of the following:

   - In the list, select the query or queries that you want to terminate, and click **Terminate Query**.
   - In the list, open a query if you want to review the query information first, and then click **Terminate Query**.

6. In the **Terminate Queries** dialog box, click **Confirm**.

Viewing Query Details

You can view details for a particular query by clicking an individual query in the table on the **Queries** page to open the **Query *ID*** view. The following list describes the information available for individual queries:

- **Query Properties**. Displays a summary of information about the query such as the query ID, the database user who ran the query, and the duration.
- **Details**. Displays the status of the query.
- **SQL**. Displays the query text in a friendly, human-readable format.
- **Query Execution Details**. Displays information about how the query was processed. This section includes both planned and actual execution data for the query. For information on using the **Query Execution Details** section, see Analyzing Query Execution (p. 189).
- **Cluster Performance During Query Execution**. Displays performance metrics from Amazon CloudWatch. For information on using the **Cluster Performance During Query Execution** section, see Viewing Cluster Performance During Query Execution (p. 192).

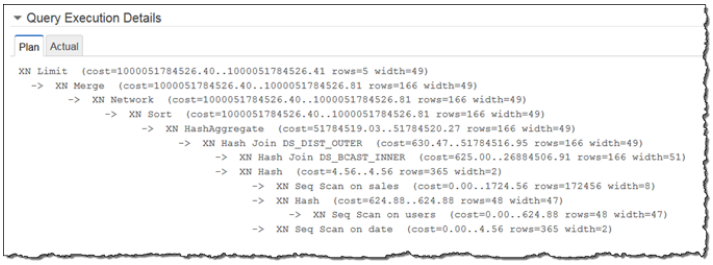
The **Query** view looks similar to the following when you open it.

Analyzing Query Execution

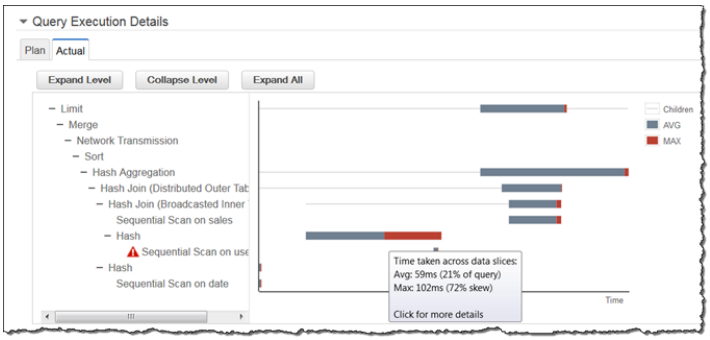
The **Query Execution Details** section of the **Query** view provides information about the way the query was processed. This section combines data from SVL_QUERY_REPORT, STL_EXPLAIN, and other system views and tables.

The **Query Execution Details** section has two tabs:

- **Plan**. This tab shows the explain plan for the query that is displayed.


Query Execution Details
Plan
Actual
XN Limit (cost=1000051784526.40..1000051784526.41 rows=5 width=49)
-> XN Merge (cost=1000051784526.40..1000051784526.81 rows=166 width=49)
-> XN Network (cost=1000051784526.40..1000051784526.81 rows=166 width=49)
-> XN Sort (cost=1000051784526.40..1000051784526.81 rows=166 width=49)
-> XN HashAggregate (cost=51784519.03..51784520.27 rows=166 width=49)
-> XN Hash Join DS_DIST_OUTER (cost=630.47..51784516.95 rows=166 width=49)
-> XN Hash Join DS_BCAST_INNER (cost=625.00..26884506.91 rows=166 width=51)
-> XN Hash (cost=4.56..4.56 rows=365 width=2)
-> XN Seq Scan on sales (cost=0.00..1724.56 rows=172456 width=8)
-> XN Hash (cost=624.88..624.88 rows=48 width=47)
-> XN Seq Scan on users (cost=0.00..624.88 rows=48 width=47)
-> XN Seq Scan on date (cost=0.00..4.56 rows=365 width=2)

- **Actual**. This tab shows the actual steps and statistics for the query that was executed. This information displays in a textual hierarchy and a visual chart. You can hover your cursor over any bar in the chart to see the **Avg** and **Max** statistics for the related step, as shown following.

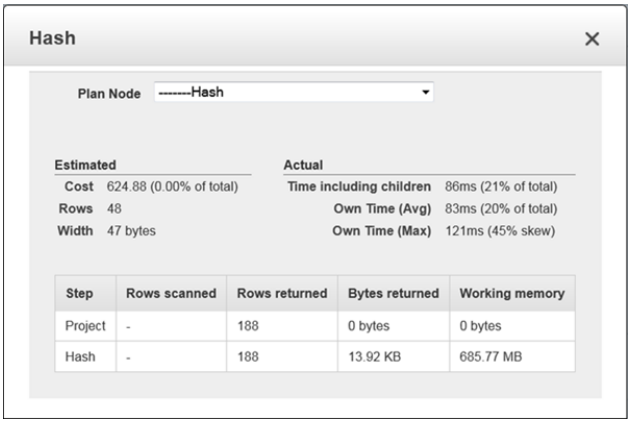


The **Avg** statistic shows the average execution time for the step across data slices, and the percentage of the total query runtime that represents. The **Max** statistic shows the longest execution time for the step on any of the data slices, and the *skew*. The skew is the difference between the average and maximum execution times for the step. You might want to investigate a step if the maximum execution time is consistently more than twice the average execution time over multiple runs of the query, and if the step also takes a significant amount of time (for example, being one of the top three steps in execution time in a large query).

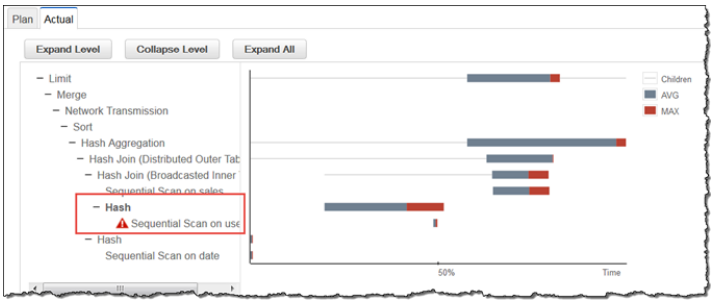
> **Note**
> When possible, you should run a query twice to see what its execution details will typically be. Compilation adds overhead to the first run of the query that is not present in subsequent runs.

To investigate high skew for a step, check the query plan for distribution steps to see what type of distribution is being performed in the query, then review your data distribution strategy to see if should be modified. For more information about Amazon Redshift data distribution, go to Choosing a Data Distribution Style in the *Amazon Redshift Database Developer Guide*.

You can click any bar in the chart to compare the data estimated from the explain plan with the actual performance of the query, as shown following.



If the query optimizer posted alerts for the query in the STL_ALERT_EVENT_LOG system table, then the plan nodes associated with the alerts are flagged with an alert icon.



The information on the **Plan** tab is analogous to running the EXPLAIN command in the database. The EXPLAIN command examines your query text, and returns the query plan. You use this information to evaluate queries, and revise them for efficiency and performance if necessary. The EXPLAIN command doesn't actually run the query.

The following example shows a query that returns the top five sellers in San Diego, based on the number of tickets sold in 2008, and the query plan for that query.

```
explain
select sellerid, username, (firstname ||' '|| lastname) as name,
city, sum(qtysold)
from sales, date, users
where sales.sellerid = users.userid
and sales.dateid = date.dateid
and year = 2008
and city = 'San Diego'
group by sellerid, username, name, city
order by 5 desc
limit 5;
```

```
QUERY PLAN
XN HashAggregate  (cost=10347367091.97..10347367466.90 rows=49990 width=24)
 ->  XN Hash Join DS_DIST_OUTER  (cost=624.88..10347366229.69 rows=172456 width=24)
       Outer Dist Key: s.buyerid
       Hash Cond: ("outer".buyerid = "inner".userid)
       ->  XN Seq Scan on sales s  (cost=0.00..1724.56 rows=172456 width=16)
       ->  XN Hash  (cost=499.90..499.90 rows=49990 width=16)
             ->  XN Seq Scan on users u  (cost=0.00..499.90 rows=49990 width=16)
```

For more information about understanding the explain plan, go to Analyzing the Explain Plan in the *Amazon Redshift Database Developer Guide*.

When you actually run the query (omitting the EXPLAIN command), the engine might find ways to optimize the query performance and change the way it processes the query. The actual performance data for the query is stored in the system views, such as SVL_QUERY_REPORT and SVL_QUERY_SUMMARY.

The Amazon Redshift console uses a combination of STL_EXPLAIN, SVL_QUERY_REPORT, and other system views and tables to present the actual query performance and compare it to the explain plan for the query. This information appears on the **Actual** tab. If you see that the explain plan and the actual query execution steps differ, you might need to perform some operations in the database, such as ANALYZE, to update statistics and make the explain plan more effective.

Additionally, sometimes the query optimizer breaks complex SQL queries into parts and creates temporary tables with the naming convention volt_tt_*guid* to process the query more efficiently. In this case, both the explain plan and the actual query execution summary apply to the last statement that was run. You can review previous query IDs to see the explain plan and actual query execution summary for each of the corresponding parts of the query.

For more information about the difference between the explain plan and system views and logs, go to Mapping the Query Plan to System Views in the Amazon Redshift Database Developer Guide in the *Amazon Redshift Database Developer Guide*.

Viewing Query Execution Details Using the Console

Use the following procedure to look at the details of query execution.

To view query execution details

1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Clusters**.
3. In the **Cluster** list, select the cluster for which you want to view query execution details.
4. Click the **Queries** tab, and open the query for which you want to view performance data.
5. Expand the **Query Execution Details** section and do the following:

   a. On the **Plan** tab, review the explain plan for the query. If you find that your explain plan differs from the actual query execution on the **Actual** tab, you might need to run ANALYZE to update statistics or perform other maintenance on the database to optimize the queries you run. For more information about query optimization, see Tuning Query Performance in the *Amazon Redshift Database Developer Guide*.

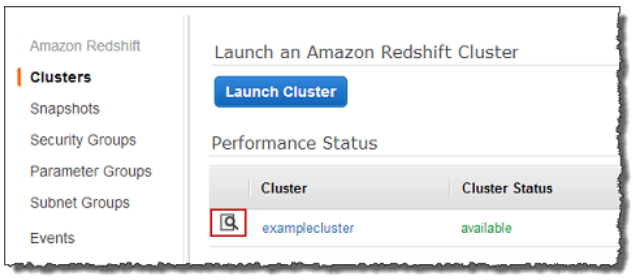
   b. On the **Actual** tab, review the performance data associated with each of the plan nodes in the query execution. You can click an individual plan node in the hierarchy to view performance data associated with that specific plan node. This data will include both the estimated and actual performance data.

Viewing Cluster Performance During Query Execution

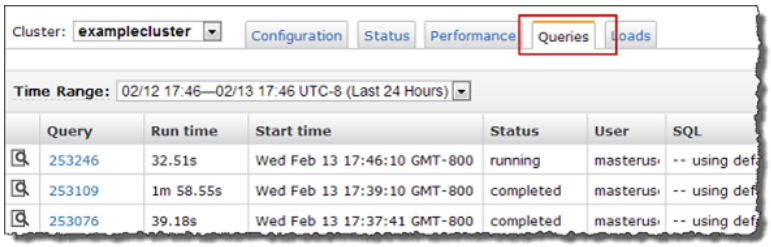
You can use the **Cluster Performance During Query Execution** section of the **Query** view to see cluster metrics during query execution to help identify poorly performing queries, look for bottleneck queries, and determine if you need to resize your cluster for your workload.

To view cluster metrics during query execution

1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Clusters**.
3. In the **Cluster** list, select the cluster for which you want to view cluster performance during query execution.

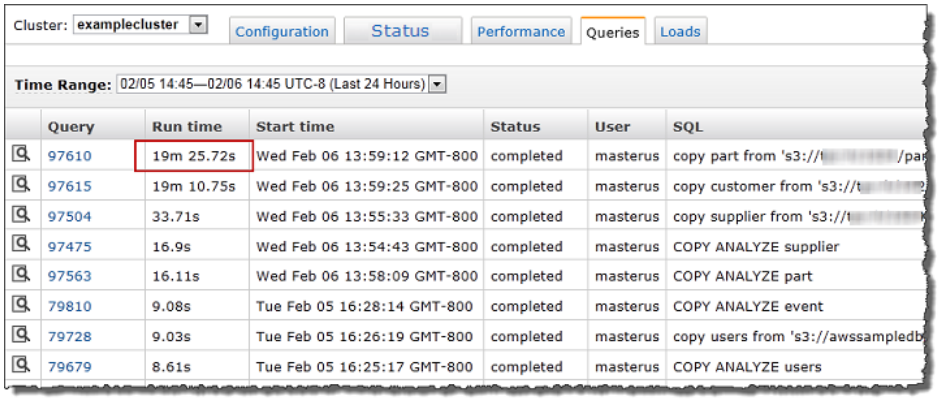


4. Click the **Queries** tab.



5. In the query list, find the query you want to work with, and click the query ID in the **Query** column.

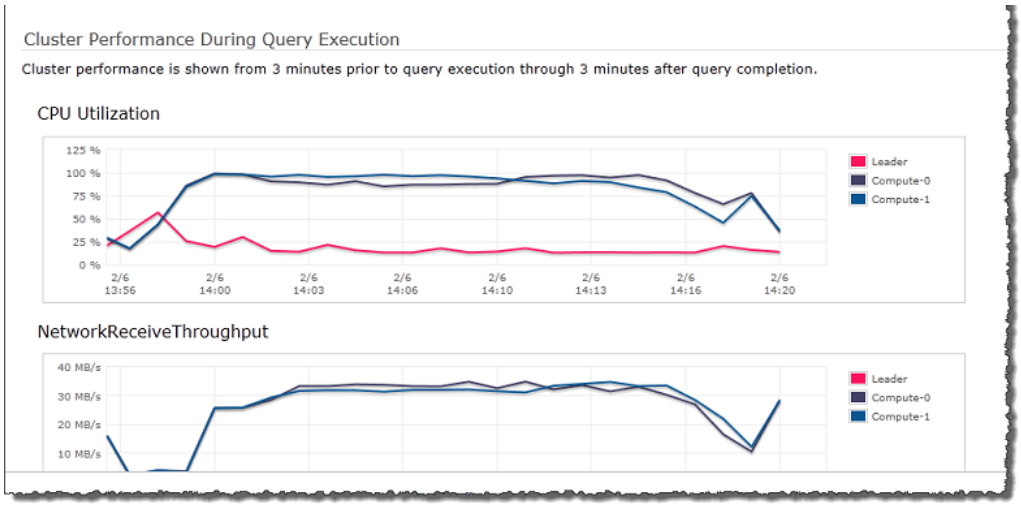
   In the following example, the queries are sorted by **Run time** to find the query with the maximum run time.



6. In the **Query** page that opens, scroll to the **Cluster Performance During Query Execution** section to view cluster metrics.

   In the following example, the `CPUUtilization` and `NetworkReceiveThroughput` metrics are displayed for the time that this query was running.

   Tip
   You can close the details of the **Query Execution Details** or **SQL** sections to manage how much information is displayed in the pane.



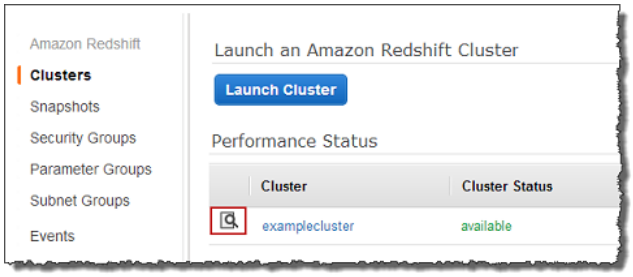
Viewing Cluster Metrics During Load Operations

When you view cluster performance during load operations, you can identify queries that are consuming resources and take action to mitigate their effect. You can terminate a load if you don't want it to run to completion.

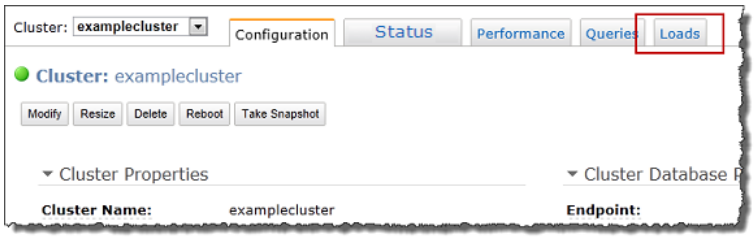
> **Note**
> The ability to terminate queries and loads in the Amazon Redshift console requires specific permission. If you select the **Amazon Redshift Read Only** AWS managed policy or create a custom policy in IAM, and you want users to be able to terminate queries and loads, make sure to add the `redshift:CancelQuerySession` action to the policy. Users who have the **Amazon Redshift Full Access** policy already have the necessary permission to terminate queries and loads. For more information about actions in IAM policies for Amazon Redshift, see IAM Policy Actions for Amazon Redshift (p. 118).

To view cluster metrics during load operations

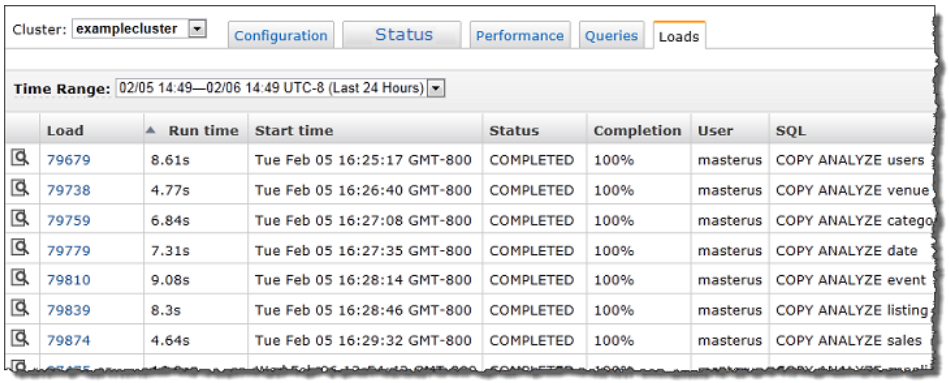
1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Clusters**.
3. In the **Cluster** list, select the cluster for which you want to view cluster performance during query execution.



4. Click the **Loads** tab.



5. In the load list, find the load operation you want to work with, and click the load ID in the **Load** column.


Cluster: examplecluster
Configuration
Status
Performance
Queries
Loads
Time Range: 02/05 14:49—02/06 14:49 UTC-8 (Last 24 Hours)
Load
Run time
Start time
Status
Completion
User
SQL
79679 8.61s Tue Feb 05 16:25:17 GMT-800 COMPLETED 100% masterus COPY ANALYZE users
79738 4.77s Tue Feb 05 16:26:40 GMT-800 COMPLETED 100% masterus COPY ANALYZE venue
79759 6.84s Tue Feb 05 16:27:08 GMT-800 COMPLETED 100% masterus COPY ANALYZE catego
79779 7.31s Tue Feb 05 16:27:35 GMT-800 COMPLETED 100% masterus COPY ANALYZE date
79810 9.08s Tue Feb 05 16:28:14 GMT-800 COMPLETED 100% masterus COPY ANALYZE event
79839 8.3s Tue Feb 05 16:28:46 GMT-800 COMPLETED 100% masterus COPY ANALYZE listing
79874 4.64s Tue Feb 05 16:29:32 GMT-800 COMPLETED 100% masterus COPY ANALYZE sales

6. In the new **Query** tab that is opened, you can view the details of the load operation.

   At this point, you can work with the **Query** tab as shown in Viewing Query Performance Data (p. 186). You can review the details of the query and see the values of cluster metrics during the load operation.

To terminate a running load

1. Sign in to the AWS Management Console, and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation pane, click **Clusters**.
3. In the **Cluster** list, click the cluster you want to open.
4. Click the **Loads** tab.
5. Do one of the following:

   - In the list, select the load or loads that you want to terminate, and click **Terminate Load**.
   - In the list, open a load if you want to review the load information first, and then click **Terminate Load**.

6. In the **Terminate Loads** dialog box, click **Confirm**.

Creating an Alarm

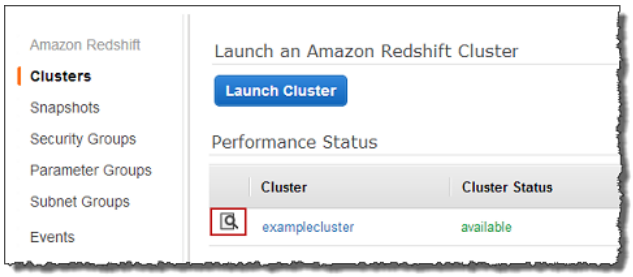
Alarms you create in the Amazon Redshift console are Amazon CloudWatch alarms. They are useful because they help you make proactive decisions about your cluster and its databases. You can set one or more alarms on any of the metrics listed in Amazon Redshift CloudWatch Metrics (p. 180). For example, setting an alarm for high `CPUUtilization` on a cluster node will help indicate when the node is over-utilized. Likewise, setting an alarm for low `CPUUtilization` on a cluster node, will help indicate when the node is underutilized.

This section explains how to create an alarm using the Amazon Redshift console. You can create an alarm using the Amazon CloudWatch console or any other way you typically work with metrics such as with the Amazon CloudWatch Command Line Interface (CLI) or one of the Amazon Software Development Kits (SDKs). To delete an alarm, you must use the Amazon CloudWatch console.

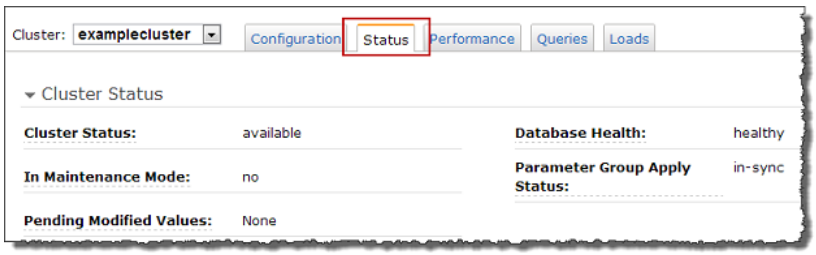
To create an alarm on a cluster metric in the Amazon Redshift console

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the left navigation, click **Clusters**.

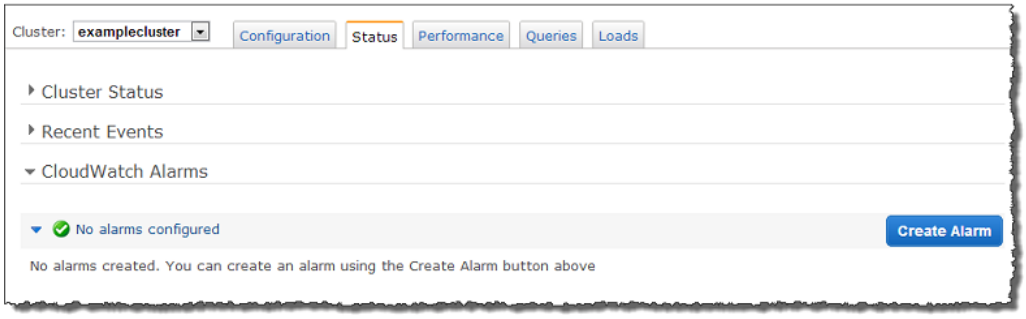
3. In the **Cluster** list, select the cluster for which you want to view cluster performance during query execution.



4. Select the **Events+Alarms** tab.



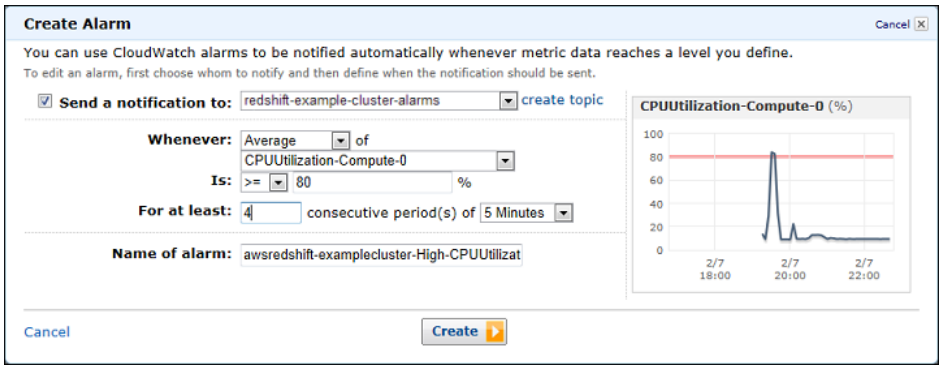
5. Click **Create Alarm**.



6. In the **Create Alarm** dialog box, configure an alarm, and click **Create**.

   Note
   The notifications that are displayed the **Send a notification to** box are your Amazon Simple Notification Service (Amazon SNS) topics. To learn more about Amazon SNS and creating topics, go to Create a Topic in the *Amazon Simple Notification Service Getting Started Guide*. If you don't have any topics in Amazon SNS, you can create a topic in the Create Alarm dialog by clicking the **create topic** link.

   The details of your alarm will vary with your circumstance. In the following example, the average CPU utilization of a node (Compute-0) has an alarm set so that if the CPU goes above 80 percent for four consecutive five minute periods, a notification is sent to the topic **redshift-example-cluster-alarms**.



7. In the list of alarms, find your new alarm.

   You may need to wait a few moments as sufficient data is collected to determine the state of the alarm as shown in the following example.

CloudWatch Alarms
1 of 1 in INSUFFICIENT DATA
Create Alarm
Below are your CloudWatch alarms for the selected resources. Click on an alarm to edit it or click 'view' to see additional options and details in Amazon CloudWatch.
View all CloudWatch alarms
State
Name
More Options
INSUFFICIENT_DATA
awsredshift-examplecluster-High-CPUUtilization-Compute-0
view

   After a few moments the state will turn to **OK**.

CloudWatch Alarms
1 of 1 in OK
Create Alarm
Below are your CloudWatch alarms for the selected resources. Click on an alarm to edit it or click 'view' to see additional options and details in Amazon CloudWatch.
View all CloudWatch alarms
State
Name
More Options
OK
awsredshift-examplecluster-High-CPUUtilization-Compute-0
view

8. (Optional) Click the **Name** of the alarm to change the configuration of the alarm or click the view link under **More Options** to go to this alarm in the Amazon CloudWatch console.

Working with Performance Metrics in the Amazon CloudWatch Console

When working with Amazon Redshift metrics in the Amazon CloudWatch console, there are couple of things you should keep in mind:

- Query and load performance data is only available in the Amazon Redshift console.
- Some Metrics in the Amazon CloudWatch have different units than those used in the Amazon Redshift console. For example, `WriteThroughput`, is displayed in GB/s (as compared to Bytes/s in Amazon CloudWatch) which is a more relevant unit for the typical storage space of a node.

When working with Amazon Redshift metrics in the Amazon CloudWatch console, command line tools, or an Amazon SDK, there are two concepts to keep in mind.

- First, you specify the metric dimension to work with. A dimension is a name-value pair that helps you to uniquely identify a metric. The dimensions for Amazon Redshift are `ClusterIdentifier` and `NodeID`. In the Amazon CloudWatch console, the `Redshift Cluster` and `Redshift Node` views are provided to easily select cluster and node-specific dimensions. For more information about dimensions, see Dimensions in the *Amazon CloudWatch Developer Guide*.
- Second, you specify the metric name, such as `ReadIOPS`.

The following table summarizes the types of Amazon Redshift metric dimensions that are available to you. All data is available in 1-minute periods at no charge.

Amazon Cloud-Watch Namespace	Dimension	Description
`AWS/Redshift`	`NodeID`	Filters requested data that is specific to the nodes of a cluster. `NodeID` will be either "Leader", "Shared", or "Compute-N" where N is 0, 1, ... for the number of nodes in the cluster. "Shared" means that the cluster has only one node, i.e. the leader node and compute node are combined.
	`ClusterIdentifier`	Filters requested data that is specific to the cluster. Metrics that are specific to clusters include `HealthStatus`, `MaintenanceMode`, and `DatabaseConnections`. In general metrics in for this dimension (e.g. `ReadIOPS`) that are also metrics of nodes represent an aggregate of the node metric data. You should take care in interpreting these metrics because they aggregate behavior of leader and compute nodes.

Working with gateway and volume metrics is similar to working with other service metrics. Many of the common tasks are outlined in the Amazon CloudWatch documentation and are listed below for your convenience:

- Listing Available Metrics
- Getting Statistics for a Metric
- Creating CloudWatch Alarms

Amazon Redshift Events

Topics

- Overview (p. 199)
- Viewing Events Using the Console (p. 199)
- Viewing Events Using the AWS SDK for Java (p. 201)
- View Events Using the Amazon Redshift CLI and API (p. 202)
- Amazon Redshift Event Notifications (p. 202)

Overview

Amazon Redshift tracks events and retains information about them for a period of several weeks in your AWS account. For each event, Amazon Redshift reports information such as the date the event occurred, a description, the event source (for example, a cluster, a parameter group, or a snapshot), and the source ID.

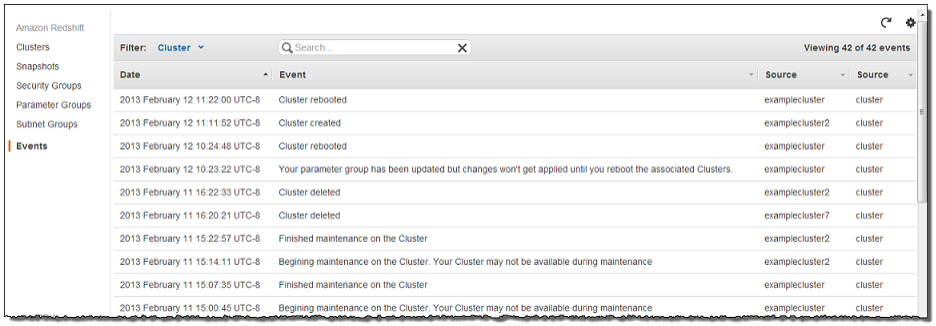
You can use the Amazon Redshift console, the Amazon Redshift API, or the AWS SDKs to obtain event information. You can obtain a list of all events, or you can apply filters—such as event duration or start and end date—to obtain events information for a specific period. You can also obtain events that were generated by a specific source type, such as cluster events or parameter group events.

You can create Amazon Redshift event notification subscriptions that specify a set of event filters. When an event occurs that matches the filter criteria, Amazon Redshift uses Amazon Simple Notification Service to actively inform you that the event has occurred.

For a list of Amazon Redshift events by source type and category, see the section called “Amazon Redshift Event Categories and Event Messages” (p. 204)

Viewing Events Using the Console

You can view events in the Amazon Redshift console by click on **Events** on the left navigation. In the list of events you can filter the results using the **Source Type** filter or a custom **Filter** that filters for text in all fields of the list. For example, if you search for "12 Dec 2012" you will match **Date** fields that contain this value.


Amazon Redshift
Clusters
Snapshots
Security Groups
Parameter Groups
Subnet Groups
Events
Filter: Cluster
Search...
Viewing 42 of 42 events
Date
Event
Source
Source
2013 February 12 11:22:00 UTC-8 Cluster rebooted examplecluster cluster
2013 February 12 11:11:52 UTC-8 Cluster created examplecluster2 cluster
2013 February 12 10:24:48 UTC-8 Cluster rebooted examplecluster cluster
2013 February 12 10:23:22 UTC-8 Your parameter group has been updated but changes won't get applied until you reboot the associated Clusters. examplecluster cluster
2013 February 11 16:22:33 UTC-8 Cluster deleted examplecluster2 cluster
2013 February 11 16:20:21 UTC-8 Cluster deleted examplecluster7 cluster
2013 February 11 15:22:57 UTC-8 Finished maintenance on the Cluster examplecluster2 cluster
2013 February 11 15:14:11 UTC-8 Begining maintenance on the Cluster. Your Cluster may not be available during maintenance examplecluster2 cluster
2013 February 11 15:07:35 UTC-8 Finished maintenance on the Cluster examplecluster cluster
2013 February 11 15:00:45 UTC-8 Begining maintenance on the Cluster. Your Cluster may not be available during maintenance examplecluster cluster

An event source type indicates what the event was about. The following source types are possible: **Cluster**, **Cluster Parameter Group**, **Cluster Security Group**, and **Snapshot**.

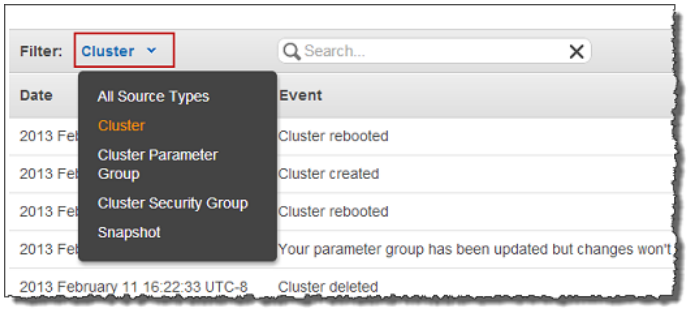
Filtering Events

Sometimes you want to find a specific category of events or events for a specific cluster. In these cases, you can filter the events displayed.

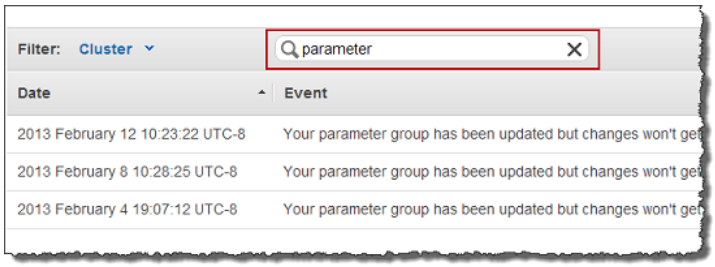
To filter events

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Events**.
3. To filter events do one of the following:

   a. To filter by event type, click **Filter Cluster** and select the source type.



   b. To filter by text that appears in the event description, type in the in the search box and the list narrows based on what you type.


Filter: Cluster
parameter
Date
Event
2013 February 12 10:23:22 UTC-8
Your parameter group has been updated but changes won't ge
2013 February 8 10:28:25 UTC-8
Your parameter group has been updated but changes won't ge
2013 February 4 19:07:12 UTC-8
Your parameter group has been updated but changes won't ge

Viewing Events Using the AWS SDK for Java

The following example lists the events for a specified cluster and specified event source type. The example shows how to use pagination.

For step-by-step instructions to run the following example, see Running Java Examples for Amazon Redshift Using Eclipse (p. 130). You need to update the code and specify a cluster identifier and event source type.

```
import java.io.IOException;
import java.util.Date;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.services.redshift.AmazonRedshiftClient;
import com.amazonaws.services.redshift.model.*;

public class ListEvents {

    public static AmazonRedshiftClient client;
   public static String clusterIdentifier = "***provide cluster identifier***";

   public static String eventSourceType = "***provide source type***"; // e.g.
 cluster-snapshot

    public static void main(String[] args) throws IOException {

        AWSCredentials credentials = new PropertiesCredentials(
                ListEvents.class
                .getResourceAsStream("AwsCredentials.properties"));

        client = new AmazonRedshiftClient(credentials);

        try {
             listEvents();
        } catch (Exception e) {
            System.err.println("Operation failed: " + e.getMessage());
        }
    }

    private static void listEvents() {
       long oneWeeksAgoMilli = (new Date()).getTime() - (7L*24L*60L*60L*1000L);

        Date oneWeekAgo = new Date();
        oneWeekAgo.setTime(oneWeeksAgoMilli);
        String marker = null;

        do {
            DescribeEventsRequest request = new DescribeEventsRequest()
            .withSourceIdentifier(clusterIdentifier)
            .withSourceType(eventSourceType)
            .withStartTime(oneWeekAgo)
            .withMaxRecords(20);
            DescribeEventsResult result = client.describeEvents(request);
            marker = result.getMarker();
            for (Event event : result.getEvents()) {
                printEvent(event);
```

```
            }
        } while (marker != null);


    }
    static void printEvent(Event event)
    {
        if (event == null)
        {
            System.out.println("\nEvent object is null.");
            return;
        }

        System.out.println("\nEvent metadata:\n");
        System.out.format("SourceID: %s\n", event.getSourceIdentifier());
        System.out.format("Type: %s\n", event.getSourceType());
        System.out.format("Message: %s\n", event.getMessage());
        System.out.format("Date: %s\n", event.getDate());
    }
}
```

View Events Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operation to manage events.

- describe-events

Amazon Redshift provides the following API to view events.

- DescribeEvents

Amazon Redshift Event Notifications

Topics

- Overview (p. 202)
- Amazon Redshift Event Categories and Event Messages (p. 204)
- Managing Event Notifications Using the Amazon Redshift Console (p. 209)
- Managing Event Notifications Using the Amazon Redshift CLI and API (p. 214)

Overview

Amazon Redshift uses the Amazon Simple Notification Service (Amazon SNS) to communicate notifications of Amazon Redshift events. You enable notifications by creating an Amazon Redshift event subscription. In the Amazon Redshift subscription, you specify a set of filters for Amazon Redshift events and an Amazon SNS topic. Whenever an event occurs that matches the filter criteria, Amazon Redshift publishes a notification message to the Amazon SNS topic. Amazon SNS then transmits the message to any Amazon SNS consumers that have an Amazon SNS subscription to the topic. The messages sent to the Amazon SNS consumers can be in any form supported by Amazon SNS for an AWS region, such as an email, a

text message, or a call to an HTTP endpoint. For example, all regions support email notifications, but SMS notifications can only be created in the US East (N. Virginia) region.

When you create an event notification subscription, you specify one or more event filters. Amazon Redshift sends notifications through the subscription any time an event occurs that matches all of the filter criteria. The filter criteria include source type (such as cluster or snapshot), source ID (such as the name of a cluster or snapshot), event category (such as Monitoring or Security), and event severity (such as INFO or ERROR).

You can easily turn off notification without deleting a subscription by setting the **Enabled** radio button to *No* in the AWS Management Console or by setting the *Enabled* parameter to `false` using the Amazon Redshift CLI or API.

Billing for Amazon Redshift event notification is through the Amazon Simple Notification Service (Amazon SNS). Amazon SNS fees apply when using event notification; for more information on Amazon SNS billing, go to Amazon Simple Notification Service Pricing.

You can also view Amazon Redshift events that have occurred by using the management console. For more information, see Amazon Redshift Events (p. 199).

Subscribing to Amazon Redshift Event Notifications

You can create an Amazon Redshift event notification subscription so you can be notified when an event occurs for a given cluster, snapshot, security group, or parameter group. The simplest way to create a subscription is with the Amazon SNS console. For information on creating an Amazon SNS topic and subscribing to it, see Getting Started with Amazon SNS.

You can create an Amazon Redshift event notification subscription so you can be notified when an event occurs for a given cluster, snapshot, security group, or parameter group. The simplest way to create a subscription is with the AWS Management Console. If you choose to create event notification subscriptions using the CLI or API, you must create an Amazon Simple Notification Service topic and subscribe to that topic with the Amazon SNS console or Amazon SNS API. You will also need to retain the Amazon Resource Name (ARN) of the topic because it is used when submitting CLI commands or API actions. For information on creating an Amazon SNS topic and subscribing to it, see Getting Started with Amazon SNS.

An Amazon Redshift event subscription can specify these event criteria:

- Source type, the values are cluster, snapshot, parameter-groups, and security-groups.
- Source ID of a resource, such as `my-cluster-1` or `my-snapshot-20130823`. The ID must be for a resource in the same region as the event subscription.
- Event category, the values are Configuration, Management, Monitoring, and Security.
- Event severity, the values are INFO or ERROR.

The event criteria can be specified independently, except that you must specify a source type before you can specify source IDs in the console. For example, you can specify an event category without having to specify a source type, source ID, or severity. While you can specify source IDs for resources that are not of the type specified in source type, no notifications will be sent for events from those resources. For example, if you specify a source type of cluster and the ID of a security group, none of the events raised by that security group would match the source type filter criteria, so no notifications would be sent for those events.

Amazon Redshift sends a notification for any event that matches all criteria specified in a subscription. Some examples of the sets of events returned:

- Subscription specifies a source type of cluster, a source ID of my-cluster-1, a category of Monitoring, and a severity of ERROR. The subscription will send notifications for only monitoring events with a severity of ERROR from my-cluster-1.
- Subscription specifies a source type of cluster, a category of Configuration, and a severity of INFO. The subscription will send notifications for configuration events with a severity of INFO from any Redshift cluster in the AWS account.
- Subscription specifies a category of Configuration, and a severity of INFO. The subscription will send notifications for configuration events with a severity of INFO from any Redshift resource in the AWS account.
- Subscription specifies a severity of ERROR. The subscription will send notifications for all events with a severity of ERROR from any Redshift resource in the AWS account.

If you delete or rename an object whose name is referenced as a source ID in an existing subscription, the subscription will remain active, but will have no events to forward from that object. If you later create a new object with the same name as is referenced in the subscription source ID, the subscription will start sending notifications for events from the new object.

Amazon Redshift publishes event notifications to an Amazon SNS topic, which is identified by its Amazon Resource Name (ARN). When you create an event subscription using the Amazon Redshift console, you can either specify an existing Amazon SNS topic, or request that the console create the topic when it creates the subscription. All Amazon Redshift event notifications sent to the Amazon SNS topic are in turn transmitted to all Amazon SNS consumers that are subscribed to that topic. Use the Amazon SNS console to make changes to the Amazon SNS topic, such as adding or removing consumer subscriptions to the topic. For more information about creating and subscribing to Amazon SNS topics, go to Getting Started with Amazon Simple Notification Service.

Amazon Redshift publishes event notifications to an Amazon SNS topic, which is identified by its Amazon Resource Name (ARN). When you create an event subscription using the Amazon Redshift console, you can either specify an existing Amazon SNS topic, or request that the console create the topic when it creates the subscription. If you use the CLI or API, you must supply the Amazon Resource Name (ARN) of an existing topic. If no Amazon SNS topic exists, you must first create one by using the Amazon SNS console or the Amazon SNS API before you create the Amazon Redshift subscription. All Amazon Redshift event notifications sent to the Amazon SNS topic are in turn transmitted to all Amazon SNS consumers that are subscribed to that topic. For more information about creating and subscribing to Amazon SNS topics, go to Getting Started with Amazon Simple Notification Service.

The following section lists all categories and events that you can be notified of. It also provides information about subscribing to and working with Amazon Redshift event subscriptions.

Amazon Redshift Event Categories and Event Messages

This section shows the event IDs and categories for each Amazon Redshift source type.

The following table shows the event category and a list of events when a cluster is the source type.

Categories and Events for the Cluster Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Configura-tion	REDSHIFT-EVENT-1000	INFO	The parameter group [parameter group name] was updated at [time]. Changes will be applied to the associated clusters when they are rebooted.

Amazon Redshift Category	Event ID	Event Severity	Description
Configuration	REDSHIFT-EVENT-1001	INFO	Your Amazon Redshift cluster [cluster name] was modified to use parameter group [parameter group name] at [time].
Configuration	REDSHIFT-EVENT-1500	ERROR	The Amazon VPC [VPC name] does not exist. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1501	ERROR	The customer subnets [subnet name] you specified for Amazon VPC [VPC name] do not exist or are invalid. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1502	ERROR	The Amazon VPC [VPC name] has no available IP addresses. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1503	ERROR	The Amazon VPC [VPC name] has no internet gateway attached to it. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Configuration	REDSHIFT-EVENT-1504	ERROR	The HSM for cluster [cluster name] is unreachable.
Configuration	REDSHIFT-EVENT-1505	ERROR	The HSM for cluster [cluster name] cannot be registered. Try a different configuration.
Management	REDSHIFT-EVENT-2000	INFO	Your Amazon Redshift cluster: [cluster name] has been created and is ready for use.
Management	REDSHIFT-EVENT-2001	INFO	Your Amazon Redshift cluster [cluster name] was deleted at [time]. A final snapshot [was / was not] saved.
Management	REDSHIFT-EVENT-2002	INFO	Your VPC security group [security group name] was updated at [time].
Management	REDSHIFT-EVENT-2003	INFO	Maintenance started on your Amazon Redshift cluster [cluster name] at [time]. The cluster may not be available during maintenance.
Management	REDSHIFT-EVENT-2004	INFO	Maintenance completed on your Amazon Redshift cluster [cluster name] at [time].
Management	REDSHIFT-EVENT-2006	INFO	A resize for your Amazon Redshift cluster [cluster name] was started at [time]. Your cluster will be in read-only mode during the resize operation.

Amazon Redshift Category	Event ID	Event Severity	Description
Management	REDSHIFT-EVENT-2007	INFO	The resize for your Amazon Redshift cluster [cluster name] is in progress. Your cluster is in read-only mode.
Management	REDSHIFT-EVENT-2008	INFO	Your restore operation to create a new Amazon Redshift cluster [cluster name] snapshot [snapshot name] was started at [time]. To monitor restore progress, please visit the AWS Management Console.
Management	REDSHIFT-EVENT-2013	INFO	Your Amazon Redshift cluster [cluster name] was renamed at [time].
Monitoring	REDSHIFT-EVENT-3000	INFO	Your Amazon Redshift cluster [cluster name] was rebooted at [time].
Monitoring	REDSHIFT-EVENT-3001	INFO	A node on your Amazon Redshift cluster: [cluster name] was automatically replaced at [time], and your cluster is operating normally.
Monitoring	REDSHIFT-EVENT-3002	INFO	The resize for your Amazon Redshift cluster [cluster name] is complete and your cluster is available for reads and writes. The resize was initiated at [time] and took [hours] hours to complete.
Monitoring	REDSHIFT-EVENT-3003	INFO	Amazon Redshift cluster [cluster name] was successfully created from snapshot [snapshot name] and is available for use.
Monitoring	REDSHIFT-EVENT-3007	INFO	Your Amazon Redshift snapshot [snapshot name] was copied successfully from [source region] to [destination region] at [time].
Monitoring	REDSHIFT-EVENT-3500	ERROR	The resize for your Amazon Redshift cluster [cluster name] failed. The resize will be automatically retried in a few minutes.
Monitoring	REDSHIFT-EVENT-3501	ERROR	Your restore operation to create Amazon Redshift cluster [cluster name] from snapshot [snapshot name] failed at [time]. Please retry your operation.
Monitoring	REDSHIFT-EVENT-3504	ERROR	The Amazon S3 bucket [bucket name] is not valid for logging for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3505	ERROR	The Amazon S3 bucket [bucket name] does not have the correct IAM policies for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3506	ERROR	The Amazon S3 bucket [bucket name] does not exist. Logging cannot continue for cluster [cluster name].
Monitoring	REDSHIFT-EVENT-3507	ERROR	The Amazon Redshift cluster [cluster name] cannot be created using EIP [IP address]. This EIP is already in use.

Amazon Redshift Category	Event ID	Event Severity	Description
Monitoring	REDSHIFT-EVENT-3508	ERROR	The Amazon Redshift cluster [cluster name] cannot be created using EIP [IP address]. The EIP cannot be found.
Monitoring	REDSHIFT-EVENT-3509	ERROR	Cross-region snapshot copy is not enabled for cluster [cluster name].
Security	REDSHIFT-EVENT-4000	INFO	Your master credentials for your Amazon Redshift cluster: [cluster name] were updated at [time].
Security	REDSHIFT-EVENT-4001	INFO	The security group [security group name] was modified at [time]. The changes will take place for all associated clusters automatically.
Security	REDSHIFT-EVENT-4500	ERROR	The security group [security group name] you provided is invalid. Your configuration changes for cluster [cluster name] were not applied. Please visit the AWS Management Console to correct the issue.
Security	REDSHIFT-EVENT-4501	ERROR	The security group [security group name] specified in Cluster Security Group [cluster security group name] could not be found. The authorization cannot be completed.

The following table shows the event category and a list of events when a parameter group is the source type.

Categories and Events for the Parameter Group Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Configuration	REDSHIFT-EVENT-1002	INFO	The parameter [parameter name] was updated from [value] to [value] at [time].
Configuration	REDSHIFT-EVENT-1003	INFO	Cluster parameter group [group name] was created.
Configuration	REDSHIFT-EVENT-1004	INFO	Cluster parameter group [group name] was deleted.
Configuration	REDSHIFT-EVENT-1005	INFO	Cluster parameter group [name] was updated at [time]. Changes will be applied to the associated clusters when they are rebooted.

The following tables shows the event category and a list of events when a security group is the source type.

Categories and Events for the Security Group Source Type

Amazon Redshift Category	Event ID	Event Severity	Description	
Security	REDSHIFT-EVENT-4002	INFO	Cluster security group [group name] was created.	
Security	REDSHIFT-EVENT-4003	INFO	Cluster security group [group name] was deleted.	
Security	REDSHIFT-EVENT-4004	INFO	Cluster security group [group name] was changed at [time]. Changes will be automatically applied to all associated clusters.	

The following tables shows the event category and a list of events when a snapshot is the source type.

Categories and Events for the Snapshot Source Type

Amazon Redshift Category	Event ID	Event Severity	Description
Management	REDSHIFT-EVENT-2009	INFO	A user snapshot [snapshot name] for Amazon Redshift Cluster [cluster name] started at [time]. To monitor snapshot progress, please visit the AWS Management Console.
Management	REDSHIFT-EVENT-2010	INFO	The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] was cancelled at [time].
Management	REDSHIFT-EVENT-2011	INFO	The user snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was deleted at [time].
Management	REDSHIFT-EVENT-2012	INFO	The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was started at [time].
Monitoring	REDSHIFT-EVENT-3004	INFO	The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] completed successfully at [time].
Monitoring	REDSHIFT-EVENT-3005	INFO	The final snapshot [name] for Amazon Redshift cluster [name] completed successfully at [time].
Monitoring	REDSHIFT-EVENT-3006	INFO	The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] was cancelled at [time].
Monitoring	REDSHIFT-EVENT-3502	ERROR	The final snapshot [snapshot name] for Amazon Redshift cluster [cluster name] failed at [time]. The team is investigating the issue. Please visit the AWS Management Console to retry the operation.

Amazon Redshift Category	Event ID	Event Severity	Description
Monitoring	REDSHIFT-EVENT-3503	ERROR	The user snapshot [snapshot name] for your Amazon Redshift cluster [cluster name] failed at [time]. The team is investigating the issue. Please visit the AWS Management Console to retry the operation.

Managing Event Notifications Using the Amazon Redshift Console

Topics

- Creating an Event Notification Subscription (p. 209)
- Listing Your Amazon Redshift Event Notification Subscriptions (p. 212)
- Modifying an Amazon Redshift Event Notification Subscription (p. 212)
- Adding a Source Identifier to an Amazon Redshift Event Notification Subscription (p. 213)
- Removing a Source Identifier from an Amazon Redshift Event Notification Subscription (p. 214)
- Deleting an Amazon Redshift Event Notification Subscription (p. 214)

You can create an Amazon Simple Notification Service (Amazon SNS) event notification subscription to send notifications when an event occurs for a given Amazon Redshift cluster, snapshot, security group, or parameter group. These notifications are sent to an SNS topic, which in turn transmits messages to any SNS consumers subscribed to the topic. The SNS messages to the consumers can be in any notification form supported by Amazon SNS for an AWS region, such as an email, a text message, or a call to an HTTP endpoint. For example, all regions support email notifications, but SMS notifications can only be created in the US East (N. Virginia) region. For more information, see Amazon Redshift Event Notifications (p. 202).

This section describes how to manage Amazon Redshift event notification subscriptions from the AWS Management Console.

Creating an Event Notification Subscription

To create an Amazon Redshift event notification subscription

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. In the Amazon Redshift Console navigation pane, click **Events**, and then click the **Subscriptions** tab.
3. In the **Subscriptions** pane, click **Create Event Subscription**.
4. In the **Create Event Subscription** dialog box, do the following:

    a. Use the **Subscription Settings** pane to specify the event filter criteria. As you select the criteria, the **Subscribed Events** list displays the Amazon Redshift events that match the criteria. Do the following:

        i. Select one or more event categories from the **Categories** box. To specify all categories, select the **Category** button. To select a subset of the categories, select the buttons for the categories to be included.

ii. Select an event severity from the **Severity** dropdown menu. If you select **Any**, events with severities of either INFO or ERROR are published. If you select **Error**, only events with a severity of ERROR are published.

iii. Select a source type from the **Source Type** dropdown menu. Only events raised by resources of that type, such as clusters or cluster parameter groups, are published by the event subscription.

iv. In the **Resources** dropdown menu, specify whether events will be published from all resources having the specified **Source Type**, or only a subset. Select **Any** to publish events from all resources of the specified type. Select **Choose Specific** if you want to select specific resources.

> **Note**
> The name of the **Resource** box changes to match the value specified in **Source Type**. For example, if you select **Cluster** in **Source Type**, the name of the **Resources** box changes to **Clusters**.

If you select **Choose Specific**, you can then specify the IDs of the specific resources whose events will be published by the event subscription. You specify the resources one at a time and add them to the event subscription. You can only specify resources that are in the same region as the event subscription. The events you have specified are listed below the **Specify IDs:** box.

A. To specify an existing resource, find the resource in the **Specify IDs:** box, and click the **+** button in the **Add** column.

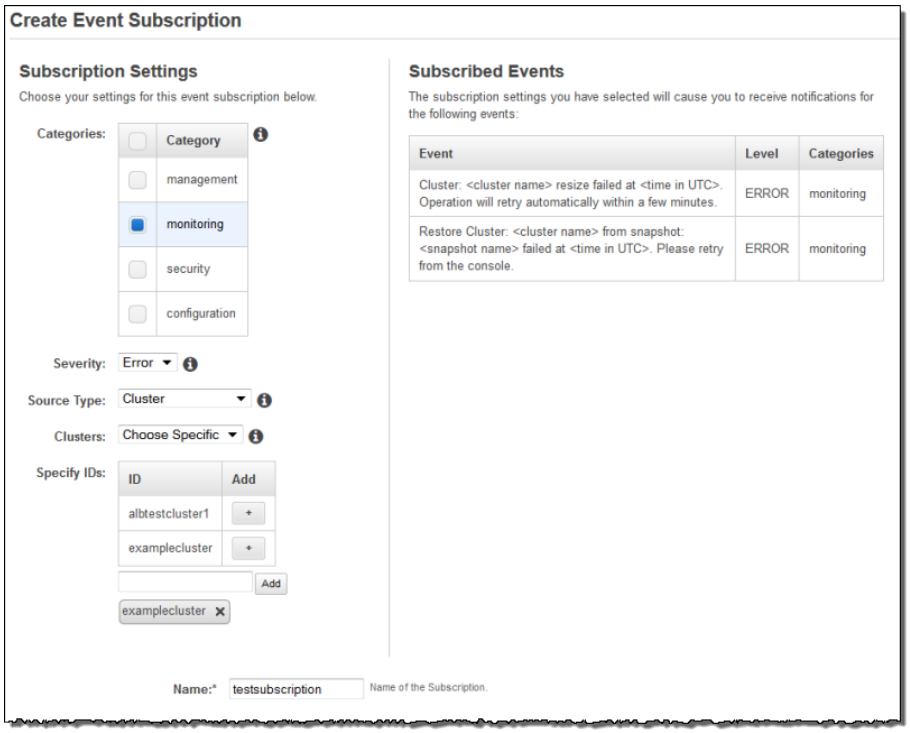
B. To specify the ID of a resource before you create it, type the ID in the box below the **Specify IDs:** box and click the **Add** button. You can do this to add resources that you plan to create later.

C. To remove a selected resource from the event subscription, click the **X** box to the right of the resource.

b. At the bottom of the pane, type a name for the event notification subscription in the **Name** text box.

c. Select **Yes** to enable the subscription. If you want to create the subscription but to not have notifications sent yet, select **No**. A confirmation message will be sent when the subscription is created, regardless of this setting.

d. Select **Next** to proceed to specifying the Amazon SNS topic.



e. Use one of three tabs to specify the Amazon SNS topic the subscription will use to publish events.

   i. To select an existing Amazon SNS topic by from a list, select the **Use Existing Topic** tab and select the topic from the list.

   ii. To specify an existing Amazon SNS topic by its Amazon Resource Name (ARN), select the **Provide Topic ARN** tab and specify the ARN in the **ARN:** box. You can find the ARN of an Amazon SNS topic by using the Amazon SNS console:

      A. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/.

      B. In the **Navigation** pane, expand **Topics**.

      C. Click the topic to be included in the Amazon Redshift event subscription.

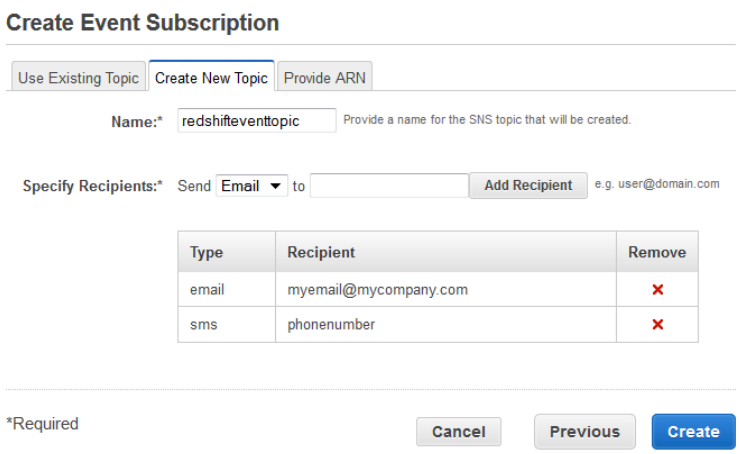
      D. In the **Topic Details** pane, copy the value of the **Topic ARN:** field.

   iii. To have the subscription create operation also create a new Amazon SNS topic, select the **Create New Topic** tab and do the following:

      A. Type a name for the topic in the **Name** text box.

      B. For each notification recipient, select the notification method in the **Send** list box, specify a valid address in the **to** box, and then click **Add Recipient**. You can only create **SMS** entries in the US East (N. Virginia) region.

      C. To remove a recipient, click the red X in the **Remove** column.



5. To create the subscription, click **Create**. To delete the definition without creating a subscription, click **Cancel**. To return to the subscription settings, click **Previous**.

Listing Your Amazon Redshift Event Notification Subscriptions

You can list your current Amazon Redshift event notification subscriptions.

To list your current Amazon Redshift event notification subscriptions

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. In the Amazon Redshift Console navigation pane, click **Events**. The **Subscriptions** tab shows all your event notification subscriptions.

Modifying an Amazon Redshift Event Notification Subscription

After you have created a subscription, you can change the subscription name, source identifier, categories, or topic ARN.

To modify an Amazon Redshift event notification subscription

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. In the Amazon Redshift Console navigation pane, click **Events**, and then click the **Subscriptions** tab.
3. In the **Subscriptions** pane, select the subscription that you want to modify, and click **Modify**.
4. In the **Modify Event Subscription** dialog box, do the following:

a. Use the **Subscription Settings** pane to change the event filter criteria. As you select the criteria, the **Subscribed Events** list displays the Amazon Redshift events that match the criteria. Do the following:

   i. Select one or more event categories from the **Categories** box. To specify all categories, select the **Category** button. To select a subset of the categories, select the buttons for the categories to be included.
   ii. Select an event severity from the **Severity** dropdown menu.
   iii. Select a source type from the **Source Type** dropdown menu.
   iv. Select the IDs of the resources from the **Source Type** dropdown menu. Only events raised by the specified resources will be published by the subscription.

b. For **Enabled**, select **Yes** to enable the subscription. Select **No** to disable the subscription.
c. Select **Next** to proceed to changing the Amazon SNS topic.
d. Use one of three tabs to change the Amazon SNS topic the subscription will use to publish events.

   i. To select an existing Amazon SNS topic by from a list, select the **Use Existing Topic** tab and select the topic from the list.
   ii. To specify an existing Amazon SNS topic by its Amazon Resource Name (ARN), select the **Provide ARN** tab and specify the ARN in the **ARN:** box.
   iii. To have the subscription modify operation also create a new Amazon SNS topic, select the **Create New Topic** tab and do the following:

      A. Type a name for the topic in the **Name** text box.
      B. For each notification recipient, select the notification method in the **Send** list box, specify a valid address in the **to** box, and then click **Add Recipient**. You can only create **SMS** entries in the US East (N. Virginia) region.
      C. To remove a recipient, click the red X in the **Remove** column.

5. To save your changes, click **Modify**. To delete your changes without modifying the subscription, click **Cancel**. To return to the subscription settings, click **Previous**.

Adding a Source Identifier to an Amazon Redshift Event Notification Subscription

You can add a source identifier (the Amazon Redshift source generating the event) to an existing subscription.

To add a source identifier to an Amazon Redshift event notification subscription

1. You can easily add or remove source identifiers using the Amazon Redshift console by selecting or deselecting them when modifying a subscription. For more information, see Modifying an Amazon Redshift Event Notification Subscription (p. 212).
2. To save your changes, click **Modify**. To delete you changes without modifying the subscription, click **Cancel**. To return to the subscription settings, click **Previous**.

Removing a Source Identifier from an Amazon Redshift Event Notification Subscription

You can remove a source identifier (the Amazon Redshift source generating the event) from a subscription if you no longer want to be notified of events for that source.

To remove a source identifier from an Amazon Redshift event notification subscription

- You can easily add or remove source identifiers using the Amazon Redshift console by selecting or deselecting them when modifying a subscription. For more information, see Modifying an Amazon Redshift Event Notification Subscription (p. 212).

Deleting an Amazon Redshift Event Notification Subscription

You can delete a subscription when you no longer need it. All subscribers to the topic will no longer receive event notifications specified by the subscription.

To delete an Amazon Redshift event notification subscription

1. Sign in to the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift/.
2. In the Amazon Redshift Console navigation pane, click **Events**, and then click the **Subscriptions** tab.
3. In the **Subscriptions** pane, click the subscription that you want to delete.
4. Click **Delete**.

Managing Event Notifications Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to manage event notifications.

- create-event-subscription
- delete-event-subscription
- describe-event-categories
- describe-event-subscriptions
- describe-events
- modify-event-subscription

You can use the following Amazon Redshift API actions to manage event notifications.

- CreateEventSubscription
- DeleteEventSubscription
- DescribeEventCategories
- DescribeEventSubscriptions
- DescribeEvents
- ModifyEventSubscription

For more information about Amazon Redshift event notifications, see Amazon Redshift Event Notifications (p. 202)

Database Audit Logging

Topics

- Overview (p. 215)
- Amazon Redshift Logs (p. 215)
- Enabling Logging (p. 217)
- Managing Log Files (p. 218)
- Troubleshooting Amazon Redshift Audit Logging (p. 220)
- Using AWS CloudTrail for Amazon Redshift (p. 221)
- Configuring Auditing Using the Console (p. 221)
- Configuring Logging by Using the Amazon Redshift CLI and API (p. 223)

Overview

Amazon Redshift logs information about connections and user activities in your database. These logs help you to monitor the database for security and troubleshooting purposes, which is a process often referred to as database auditing. The logs are stored in the Amazon Simple Storage Service (Amazon S3) buckets for convenient access with data security features for users who are responsible for monitoring activities in the database.

Amazon Redshift Logs

Amazon Redshift logs information in the following log files:

- *Connection log* — logs authentication attempts, and connections and disconnections.
- *User log* — logs information about changes to database user definitions.
- *User activity log* — logs each query before it is run on the database.

The connection and user logs are useful primarily for security purposes. You can use the connection log to monitor information about the users who are connecting to the database and the related connection information, such as their IP address, when they made the request, what type of authentication they used, and so on. You can use the user log to monitor changes to the definitions of database users.

The user activity log is useful primarily for troubleshooting purposes. It tracks information about the types of queries that both the users and the system perform in the database.

The connection log and user log both correspond to information that is stored in the system tables in your database. You can use the system tables to obtain the same information, but the log files provide an easier mechanism for retrieval and review. The log files rely on Amazon S3 permissions rather than database permissions to perform queries against the tables. Additionally, by viewing the information in log files rather than querying the system tables, you reduce any impact of interacting with the database.

Connection Log

Logs authentication attempts, and connections and disconnections. The following table describes the information in the connection log.

Column name	Description
event	Connection or authentication event.
recordtime	Time the event occurred.
remotehost	Name or IP address of remote host.
remoteport	Port number for remote host.
pid	Process ID associated with the statement.
dbname	Database name.
username	User name.
authmethod	Authentication method.
duration	Duration of connection in microseconds.

User Log

Records details for the following changes to a database user:

- Create user
- Drop user
- Alter user (rename)
- Alter user (alter properties)

Column name	Description
userid	ID of the user who generated the entry.
username	User name of the user affected by the change.
oldusername	For a rename action, the original user name. For any other action, this field is empty.

Column name	Description
action	Action that occurred. Valid values: • Alter • Create • Drop • Rename
usecreatedb	If true (1), indicates that the user has create database privileges.
usesuper	If true (1), indicates that the user is a superuser.
usecatupd	If true (1), indicates that the user can update system catalogs.
valuntil	Password expiration date.
pid	Process ID.
xid	Transaction ID.
recordtime	Time in UTC that the query started.

User Activity Log

Logs each query before it is run on the database.

Column name	Description
recordtime	Time the event occurred.
db	Database name.
user	User name.
pid	Process ID associated with the statement.
userid	User ID.
xid	Transaction ID.
query	A prefix of LOG: followed by the text of the query, including newlines.

Enabling Logging

Audit logging is not enabled by default in Amazon Redshift. When you enable logging on your cluster, Amazon Redshift creates and uploads logs to Amazon S3 that capture data from the creation of the cluster to the present time. Each logging update is a continuation of the information that was already logged.

Note
Audit logging to Amazon S3 is an optional, manual process. When you enable logging on your cluster, you are enabling logging to Amazon S3 only. Logging to system tables is not optional and happens automatically for the cluster. For more information about logging to system tables, see System Tables Reference in the Amazon Redshift Database Developer Guide.

The connection log, user log, and user activity log are enabled together by using the AWS Management Console, the Amazon Redshift API Reference, or the AWS Command Line Interface (AWS CLI). For the user activity log, you must also enable the *enable_user_activity_logging* database parameter. If you enable only the audit logging feature, but not the associated parameter, the database audit logs will log information for only the connection log and user log, but not for the user activity log. The *enable_user_activity_logging* parameter is disabled (`false`) by default, but you can set it to `true` to enable the user activity log. For more information, see Amazon Redshift Parameter Groups (p. 56).

Managing Log Files

The number and size of Amazon Redshift log files in Amazon S3 will depend heavily on the activity in your cluster. At a minimum, Amazon Redshift uploads logging information on an hourly basis. If you have an active cluster that is generating large logs, Amazon Redshift may upload the log files more frequently. You might have a series of log files for the same type of activity, such as having multiple connection logs within the same hour.

Because Amazon Redshift uses Amazon S3 to store logs, you will incur charges for the storage that you use in Amazon S3. Before you configure logging, you should have a plan for how long you need to store the log files, and determine when they can either be deleted or archived based on your auditing needs. The plan that you create depends heavily on the type of data that you store, such as data subject to compliance or regulatory requirements. For more information about Amazon S3 pricing, go to Amazon Simple Storage Service (S3) Pricing.

Bucket Permissions for Amazon Redshift Audit Logging

When you enable logging, Amazon Redshift collects logging information and uploads it to log files stored in Amazon S3. You can use an existing bucket or a new bucket. Amazon Redshift requires the following IAM permissions to the bucket:

- *s3:GetBucketAcl* The service requires read permissions to the Amazon S3 bucket so it can identify the bucket owner.
- *s3:PutObject* The service requires put object permissions to upload the logs. Each time logs are uploaded, the service determines whether the current bucket owner matches the bucket owner at the time logging was enabled. If these owners do not match, logging is still enabled but no log files can be uploaded until you select a different bucket.

If you want to use a new bucket, and have Amazon Redshift create it for you as part of the configuration process, the correct permissions will be applied to the bucket. However, if you create your own bucket in Amazon S3 or use an existing bucket, you need to add a bucket policy that includes the bucket name, and the Amazon Redshift Account ID that corresponds to your region from the following table:

Region	Account ID
US East (N. Virginia) region	193672423079
US West (Oregon) region	902366379725
EU (Frankfurt) region	053454850223
EU (Ireland) region	210876761215
Asia Pacific (Tokyo) region	404641285394

Region	Account ID
Asia Pacific (Singapore) region	361669875840
Asia Pacific (Sydney) region	762762565011

The bucket policy uses the following format, where *BucketName* and *AccountId* are placeholders for your own values:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "Put bucket policy needed for audit logging",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::AccountId:user/logs"
   },
   "Action": "s3:PutObject",
   "Resource": "arn:aws:s3:::BucketName/*"
  },
  {
   "Sid": "Get bucket policy needed for audit logging ",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::AccountID:user/logs"
   },
   "Action": "s3:GetBucketAcl",
   "Resource": "arn:aws:s3:::BucketName"
  }
 ]
}
```

The following example is a bucket policy for the US East (N. Virginia) region and bucket named AuditLogs.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "Put bucket policy needed for audit logging",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::193672423079:user/logs"
   },
   "Action": "s3:PutObject",
   "Resource": "arn:aws:s3:::AuditLogs/*"
  },
  {
   "Sid": "Get bucket policy needed for audit logging ",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::193672423079:user/logs"
   },
```

```
   "Action": "s3:GetBucketAcl",
   "Resource": "arn:aws:s3:::AuditLogs"
  }
 ]
}
```

For more information about creating Amazon S3 buckets and adding bucket policies, go to Creating a Bucket and Editing Bucket Permissions in the Amazon Simple Storage Service Console User Guide.

Bucket Structure for Amazon Redshift Audit Logging

By default, Amazon Redshift organizes the log files in the Amazon S3 bucket by using the following bucket and object structure:
AwsLogs/*AccountID*/*ServiceName*/*Region*/*Year*/*Month*/*Day*/*AccountID_ServiceName_Region_ClusterName_LogType_Timestamp.gz*

For example:
AwsLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz

If you provide an Amazon S3 key prefix, the prefix is placed at the start of the key.

For example, if you specify a prefix of myprefix:
myprefix/AwsLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz

The Amazon S3 key prefix cannot exceed 512 characters. It cannot contain spaces ( ), double quotation marks (“), single quotation marks (‘), a backslash (\). There are also a number of special characters and control characters that are not allowed. The hexadecimal codes for these characters are:

- x00 to x20
- x22
- x27
- x5c
- x7f or larger

Troubleshooting Amazon Redshift Audit Logging

Amazon Redshift audit logging can be interrupted for the following reasons:

- Amazon Redshift does not have permission to upload logs to the Amazon S3 bucket. Verify that the bucket is configured with the correct IAM policy. For more information, see Bucket Permissions for Amazon Redshift Audit Logging (p. 218).
- The bucket owner changed. When Amazon Redshift uploads logs, it verifies that the bucket owner is the same as when logging was enabled. If the bucket owner has changed, Amazon Redshift cannot upload logs until you configure another bucket to use for audit logging. For more information, see Modifying the Bucket for Audit Logging (p. 222).
- The bucket cannot be found. If the bucket is deleted in Amazon S3, Amazon Redshift cannot upload logs. You either need to recreate the bucket or configure Amazon Redshift to upload logs to a different bucket. For more information, see Modifying the Bucket for Audit Logging (p. 222).

Using AWS CloudTrail for Amazon Redshift

This service supports AWS CloudTrail, which is a service that records AWS calls for your AWS account and delivers log files to an Amazon S3 bucket. By using information collected by CloudTrail, you can determine what requests were successfully made to AWS services, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to turn it on and find your log files, see the AWS CloudTrail User Guide.

CloudTrail can be used independently from or in addition to Amazon Redshift database audit logging.

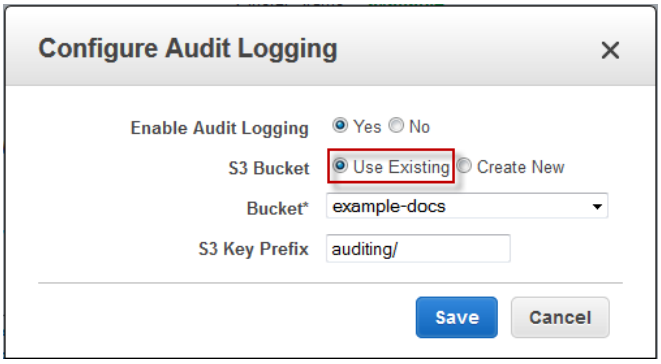
Configuring Auditing Using the Console

You can configure Amazon Redshift to create audit log files and store them in S3.

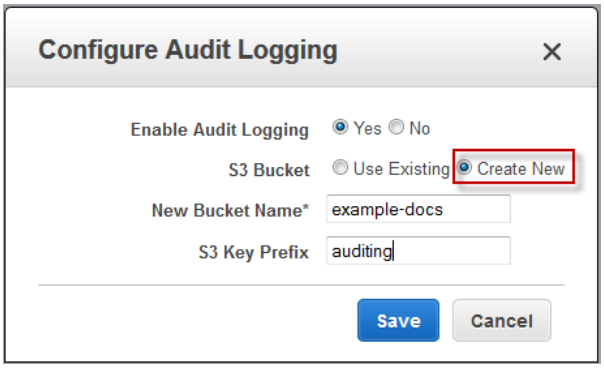
Enabling Audit Logging Using the Console

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**.
3. In the list, click the cluster for which you want to enable logging.
4. In the cluster details page, click **Database**, and then click **Configure Audit Logging**.
5. In the **Configure Audit Logging** dialog box, in the **Enable Audit Logging** box, click **Yes**.
6. For **S3 Bucket**, do one of the following:

   - If you already have an S3 bucket that you want to use, select **Use Existing** and then select the bucket from the **Bucket** list.

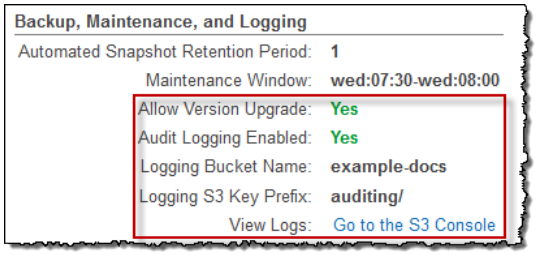


   - If you need a new S3 bucket, select **Create New**, and in the **New Bucket Name** box, type a name.



7. Optionally, in the **S3 Key Prefix** box, type a prefix to add to the S3 bucket.
8. Click **Save**.

After you configure audit logging, the **Cluster** details page updates to display information about the logging configuration.


Backup, Maintenance, and Logging
Automated Snapshot Retention Period: 1
Maintenance Window: wed:07:30-wed:08:00
Allow Version Upgrade: Yes
Audit Logging Enabled: Yes
Logging Bucket Name: example-docs
Logging S3 Key Prefix: auditing/
View Logs: Go to the S3 Console

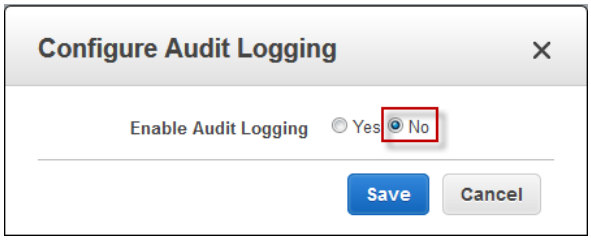
On the **Cluster** details page, under **Backup, Maintenance, and Logging**, click **Go to the S3 console** to navigate to the bucket.

Modifying the Bucket for Audit Logging

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**.
3. In the list, click the cluster for which you want to modify the bucket used for audit logging.
4. In the cluster details page, click **Database**, and then click **Configure Audit Logging**.
5. For **S3 Bucket**, select an existing bucket or create a new bucket.
6. Optionally, in the **S3 Key Prefix** box, type a prefix to add to the S3 bucket.
7. Click **Save**.

Disabling Audit Logging Using the Console

1. Sign into the AWS Management Console and open the Amazon Redshift console at https://console.aws.amazon.com/redshift.
2. In the navigation pane, click **Clusters**.
3. In the list, click the cluster for which you want to disable logging.
4. In the cluster details page, click **Database**, and then click **Configure Audit Logging**.
5. In the **Configure Audit Logging** dialog box, in the **Enable Audit Logging** box, click **No**.



6. Click **Save**.

Configuring Logging by Using the Amazon Redshift CLI and API

You can use the following Amazon Redshift CLI operations to configure audit logging.

- describe-logging-status
- disable-logging
- enable-logging

You can use the following Amazon Redshift API actions to configure audit logging.

- DescribeLoggingStatus
- DisableLogging
- EnableLogging

Tutorial: Resizing Clusters in Amazon Redshift

Topics

- Overview (p. 224)
- Resize Operation Overview (p. 224)
- Snapshot, Restore, and Resize Operation Overview (p. 225)
- Tutorial: Using the Resize Operation to Resize a Cluster (p. 226)
- Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster (p. 228)

Overview

As your data warehousing capacity and performance needs change or grow, you can resize your cluster to make the best use of the computing and storage options that Amazon Redshift provides. You can scale the cluster in or out by changing the number of nodes. Or, you can scale the cluster up or down by specifying a different node type. You can resize your cluster by using one of the following approaches:

- Use the resize operation with an existing cluster.
- Use the snapshot and restore operations to make a copy of an existing cluster. Then, resize the new cluster.

Both the resize approach and the snapshot and restore approach copy user tables and data to the new cluster; they do not do anything with system tables and data. If you have enabled audit logging in your source cluster, you'll be able to continue to access the logs in Amazon Simple Storage Service (Amazon S3) even after you delete the source cluster. You can keep or delete these logs as your data policies specify.

Resize Operation Overview

The resize operation is the preferred method to resize your cluster because it is the simplest method. With the resize operation, your data is copied in parallel from the compute node or nodes in your source cluster to the compute node or nodes in the target cluster. The time that it takes to resize depends on

the amount of data and the number of nodes in the smaller cluster. It can take anywhere from a couple of hours to a couple of days.

When you start the resize operation, Amazon Redshift puts the existing cluster into read-only mode until the resize finishes. During this time, you can only run queries that read from the database; you cannot run any queries that write to the database, including read-write queries. For more information, see Write and read-write operations in the *Amazon Redshift Database Developer Guide.*

> **Note**
> If you would like to resize with minimal production impact, you can use the following section, Snapshot, Restore, and Resize Operation Overview (p. 225), to create a copy of your cluster, resize the copy, and then switch the connection endpoint to the resized cluster when the resize is complete.

After Amazon Redshift puts the source cluster into read-only mode, it provisions a new cluster, the target cluster, using the information that you specify for the node type, cluster type, and number of nodes. Then, Amazon Redshift copies the data from the source cluster to the target cluster. When this is complete, all connections switch to use the target cluster. If you have any queries in progress at the time this switch happens, your connection will be lost and you must restart the query on the target cluster. You can view the resize progress on the cluster's **Status** tab on the Amazon Redshift console.

Amazon Redshift does not sort tables during a resize operation, so the existing sort order is maintained. When you resize a cluster, Amazon Redshift distributes the database tables to the new nodes based on their distribution styles and runs an ANALYZE command to update statistics. Rows that are marked for deletion are not transferred, so you will only need to run a VACUUM command if your tables need to be resorted. For more information, see Vacuuming tables in the *Amazon Redshift Database Developer Guide.*

To walk through the process of resizing an Amazon Redshift cluster using the resize operation, see Tutorial: Using the Resize Operation to Resize a Cluster (p. 226).

Snapshot, Restore, and Resize Operation Overview

As described in the preceding section, the time it takes to resize a cluster with the resize operation depends heavily on the amount of data in the cluster. Because you cannot perform write or read-write operations in the database during the resize, you should determine whether you want to use the resize operation or an alternate method that reduces the amount of time that the cluster is in read-only mode.

If you require near-constant write access to your Amazon Redshift cluster, you can use the snapshot and restore operations described in the following section. This approach requires that any data that is written to the source cluster after the snapshot is taken must be copied manually to the target cluster after the switch. Depending on how long the copy takes, you might need to repeat this several times until you have the same data in both clusters and can make the switch to the target cluster. This process might have a negative impact on existing queries until the full set of data is available in the target cluster, but it does minimize the amount of time that you cannot write to the database.

The snapshot, restore, and resize approach uses the following process:

1. Take a snapshot of your existing cluster. The existing cluster is the source cluster.
2. Make note of the time the snapshot was taken so that you can later identify the point at which you'll need to rerun extract, transact, load (ETL) processes to load any post-snapshot data into the target database.
3. Restore the snapshot into a new cluster. This new cluster is the target cluster. Verify that the sample data exists in the target cluster.

4. Resize the target cluster. Select the new node type, number of nodes, and other settings for the target cluster.
5. Review the loads from your ETL processes that occurred after you took a snapshot of the source cluster. You'll need to reload the same data in the same order into the target cluster. If you have ongoing data loads, you'll need to repeat this process several times until the data is the same in both the source and target clusters.
6. Stop all queries running on the source cluster. To do this, you can reboot the cluster, or you can log in as a super user and use the PG_CANCEL_BACKEND and the PG_TERMINATE_BACKEND commands. Rebooting the cluster is the easiest way to make sure that the cluster is unavailable.
7. Rename the source cluster. For example, rename it from `examplecluster` to `examplecluster-source`.
8. Rename the target cluster to use the name of the source cluster prior to the rename. For example, rename the target cluster from preceding to `examplecluster`. From this point on, any applications that use the endpoint containing `examplecluster` will be connecting to the target cluster.
9. Delete the source cluster after you switch to the target cluster, and verify that all processes work as expected.

Alternatively, you can rename the source and target clusters before reloading data into the target cluster if you do not have a requirement that any dependent systems and reports be immediately up-to-date with those for the target cluster. In this case, step 6 would be moved to the end of the process described preceding.

The rename process is only required if you want applications to continue using the same endpoint to connect to the cluster. If you do not require this, you can instead update any applications that connect to the cluster to use the endpoint of the target cluster without renaming the cluster.

There are a couple of benefits to reusing a cluster name. First, you do not need to update application connection strings because the endpoint does not change, even though the underlying cluster changes. Second, related items such as Amazon CloudWatch alarms and Amazon Simple Notification Service (Amazon SNS) notifications are tied to the cluster name, so you can continue using the same alarms and notifications that you've set up for the cluster. This continued use is primarily a concern in production environments where you want to have the freedom to resize the cluster without having to reconfigure related items, such as alarms and notifications.

To walk through the process of resizing an Amazon Redshift cluster using the snapshot, restore, and resize operations, see Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster (p. 228).

Tutorial: Using the Resize Operation to Resize a Cluster

This section walks you through the process of resizing a cluster by using the resize operation in Amazon Redshift. In this example, you'll scale your cluster out by resizing from a single node cluster to a multinode cluster.

Complete this tutorial by performing the steps in the following:

- Prerequisites (p. 227)
- Step 1: Resize the Cluster (p. 227)
- Step 2: Delete the Sample Cluster (p. 228)

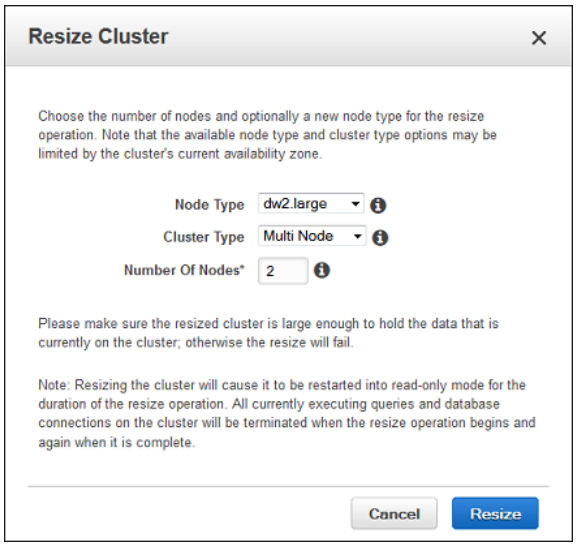
Prerequisites

Before you start this tutorial, make sure that you have the following prerequisites:

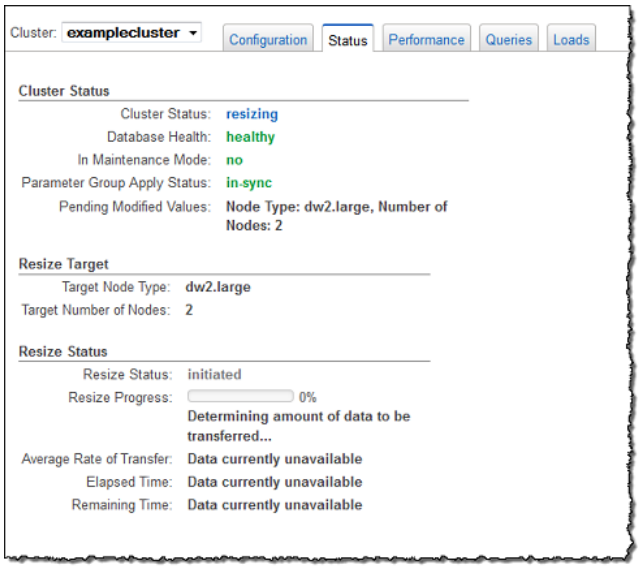
- A sample cluster. In this example, you'll start with the sample cluster that you created in the Amazon Redshift Getting Started exercise. If you don't have a sample cluster to use for this tutorial, complete the Getting Started exercise to create one and then return to this tutorial.

Step 1: Resize the Cluster

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from the Amazon Redshift Getting Started exercise, click **examplecluster**.
3. On the **Configuration** tab of the **Cluster** details page, click **Resize** in the **Cluster** list.
4. In the **Resize Cluster** window, select the following values:
   - **Node Type**: **dw2.large**.
   - **Cluster Type**: **Multi Node**.
   - **Number of Nodes**: **2**.


Resize Cluster
Choose the number of nodes and optionally a new node type for the resize operation. Note that the available node type and cluster type options may be limited by the cluster's current availability zone.
Node Type
dw2.large
Cluster Type
Multi Node
Number Of Nodes*
2
Please make sure the resized cluster is large enough to hold the data that is currently on the cluster; otherwise the resize will fail.
Note: Resizing the cluster will cause it to be restarted into read-only mode for the duration of the resize operation. All currently executing queries and database connections on the cluster will be terminated when the resize operation begins and again when it is complete.
Cancel
Resize

5. Click **Resize**.
6. Click **Status**, and review the resize status information to see the resize progress.



Step 2: Delete the Sample Cluster

After you are sure that you no longer need the sample cluster, you can delete it. In a production environment, whether you decide to keep a final snapshot depends on your data policies. In this tutorial, you'll delete the cluster without a final snapshot because you are using sample data.

> **Important**
> You are charged for any clusters until they are deleted.

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster names from this tutorial, click **examplecluster**.
3. On the **Configuration** tab of the **Cluster** details page, click **Delete** in the **Cluster** list.
4. In the **Delete Cluster** window, click **No** for **Create final snapshot**, and then click **Delete**.

Tutorial: Using the Snapshot, Restore, and Resize Operations to Resize a Cluster

This section walks you through the process of using the snapshot and restore operations as part of a resize process for an Amazon Redshift cluster. This process is an advanced one that is useful primarily in environments where you are unable or do not want to stop write and read-write operations in the database for the period of time it takes to resize your cluster. If you are unsure how long your cluster takes to resize, you can use this procedure to take a snapshot, restore it into a new cluster, and then resize it to get an estimate. This section takes that process further by switching from the source to the target cluster after the resize of the target cluster completes.

> **Important**
> You are charged for any clusters until they are deleted.

Complete this tutorial by performing the steps in the following:

- Prerequisites (p. 229)
- Step 1: Take a Snapshot (p. 229)

- Step 2: Restore the Snapshot into the Target Cluster (p. 230)
- Step 3: Verify Data in the Target Cluster (p. 231)
- Step 4: Resize the Target Cluster (p. 232)
- Step 5: Copy Post-Snapshot Data from the Source to the Target Cluster (p. 233)
- Step 6: Rename the Source and Target Clusters (p. 234)
- Step 7: Delete the Source Cluster (p. 235)
- Step 8: Clean Up Your Environment (p. 236)

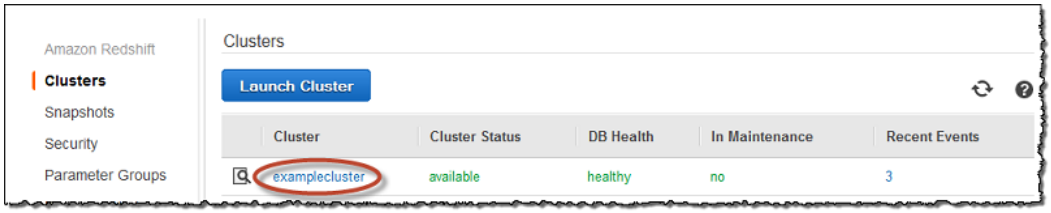
Prerequisites

Before you start this tutorial, make sure that you have the following prerequisites:

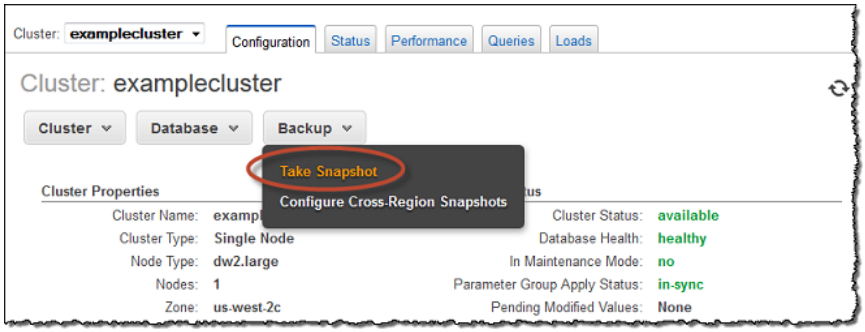
- A sample cluster. In this example, you'll start with the sample cluster that you created in the Amazon Redshift Getting Started exercise. If you don't have a sample cluster to use for this tutorial, complete the Getting Started exercise to create one and then return to this tutorial.
- A SQL client tool or application to connect to the cluster. This tutorial uses SQL Workbench/J, which you installed if you performed the steps in the Amazon Redshift Getting Started exercise. If you do not have SQL Workbench/J or another SQL client tool, see Connect to Your Cluster by Using SQL Workbench/J (p. 162).
- Sample data. In this tutorial, you'll take a snapshot of your cluster, and then perform some write queries in the database that cause a difference between the data in the source cluster and the new cluster where you will restore the snapshot. Before you begin this tutorial, load your cluster with the sample data from Amazon S3 as described in the Amazon Redshift Getting Started exercise.

Step 1: Take a Snapshot

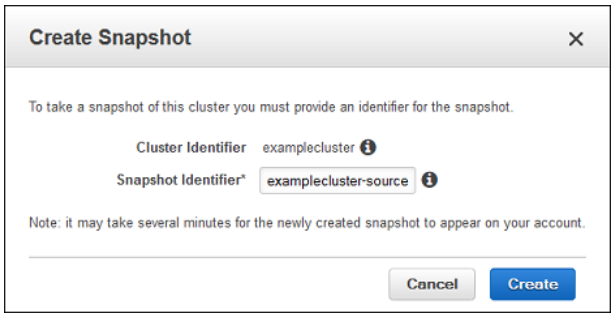
1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from the Amazon Redshift Getting Started exercise, click **examplecluster**.



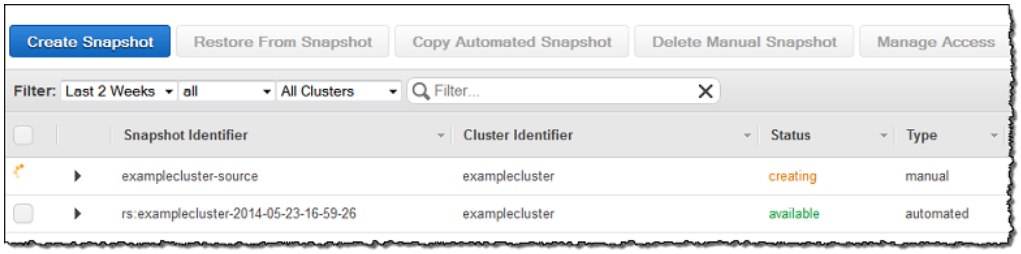
3. On the **Configuration** tab of the **Cluster** details page, click **Take Snapshot** in the **Backup** list.



4. In the **Create Snapshot** window, type *examplecluster-source* in the **Snapshot Identifier** box, and then click **Create**.

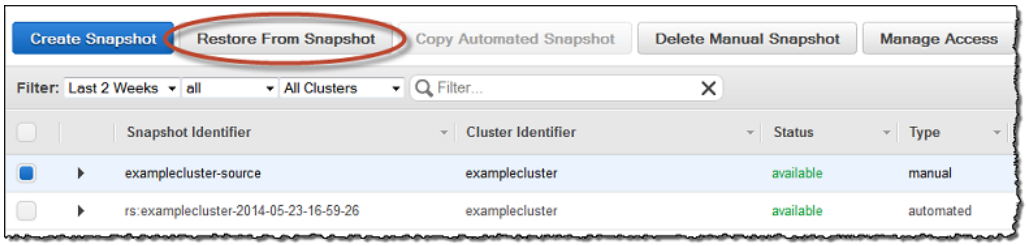

Create Snapshot
To take a snapshot of this cluster you must provide an identifier for the snapshot.
Cluster Identifier examplecluster
Snapshot Identifier* examplecluster-source
Note: it may take several minutes for the newly created snapshot to appear on your account.
Cancel
Create

5. In the navigation pane, click **Snapshots** and verify that a new manual snapshot is being created. The snapshot status will be **creating**.



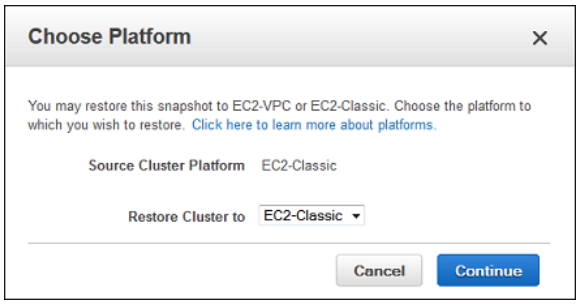
Step 2: Restore the Snapshot into the Target Cluster

1. In the navigation pane, click **Snapshots**, and then select the **examplecluster-source** snapshot.
2. Click **Restore From Snapshot**.



3. In the **Choose Platform** window, select the platform you want to restore the cluster into. If your account and region continue to support the EC2-Classic platform, choose **EC2-Classic**. Otherwise, choose **EC2-VPC**. Then, click **Continue**.

   Note
   If you choose EC2-VPC, you must have a cluster subnet group. For more information, see Creating a Cluster Subnet Group (p. 37).


Choose Platform
You may restore this snapshot to EC2-VPC or EC2-Classic. Choose the platform to which you wish to restore. Click here to learn more about platforms.
Source Cluster Platform EC2-Classic
Restore Cluster to EC2-Classic
Cancel
Continue

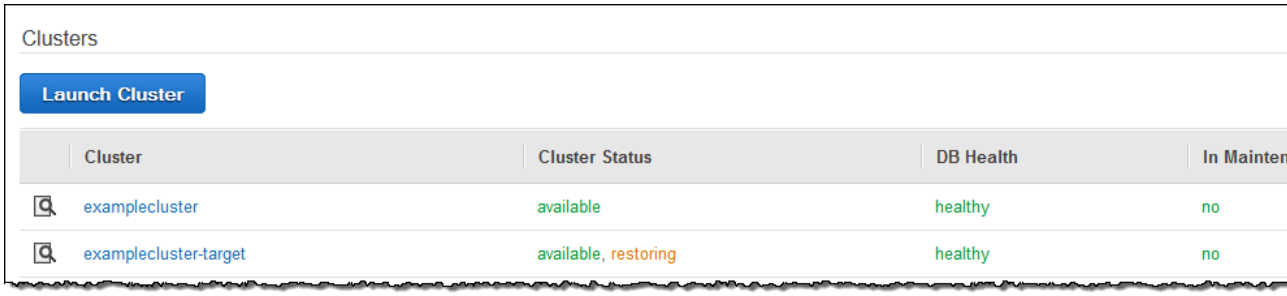
4. In the **Restore Cluster From Snapshot** window, do the following:

- **Snapshot Identifier**: check the snapshot name, **examplecluster-source**.
- **Cluster Identifier**: type **examplecluster-target**.
- **Port**: leave the port number as is.
- **Allow Version Upgrade**: leave this option as **Yes**.
- **Availability Zone**: select an Availability Zone.
- **Cluster Parameter Group**: select a parameter group to use.
- **Cluster Security Group**: select a security group or groups to use.

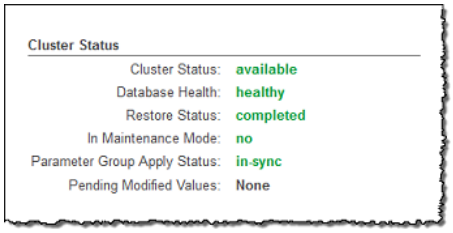
5. In the navigation pane, click **Clusters**. A new cluster, **examplecluster-target**, will be created from the source cluster's snapshot.

   First, the target cluster is created. The **Cluster Status** value is **creating, restoring** at this point.

   After the target cluster is created, the **Cluster Status** value changes to **available, restoring**.



6. Click **examplecluster-target** to open it. The **Cluster Status** value should display **available**, and the **Restore Status** should display **completed**.



Step 3: Verify Data in the Target Cluster

After the restore operation completes, you can verify that the data in the target cluster meets your expectation of the data that you had in the snapshot from the source. You can use a SQL client tool to connect to the target cluster and run a query to validate the data in the new cluster. For example, you can run the same queries that you ran in the Amazon Redshift Getting Started exercise:

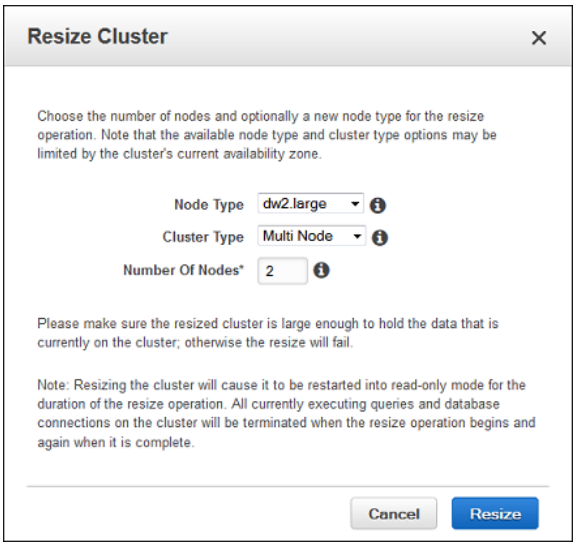
```
-- Get definition for the sales table.
SELECT *
```

```
FROM pg_table_def
WHERE tablename = 'sales';

-- Find total sales on a given calendar date.
SELECT sum(qtysold)
FROM   sales, date
WHERE  sales.dateid = date.dateid
AND    caldate = '2008-01-05';

-- Find top 10 buyers by quantity.
SELECT firstname, lastname, total_quantity
FROM   (SELECT buyerid, sum(qtysold) total_quantity
        FROM  sales
        GROUP BY buyerid
        ORDER BY total_quantity desc limit 10) Q, users
WHERE Q.buyerid = userid
ORDER BY Q.total_quantity desc;

-- Find events in the 99.9 percentile in terms of all-time gross sales.
SELECT eventname, total_price
FROM  (SELECT eventid, total_price, ntile(1000) over(order by total_price desc)
 as percentile
       FROM (SELECT eventid, sum(pricepaid) total_price
             FROM   sales
             GROUP BY eventid)) Q, event E
       WHERE Q.eventid = E.eventid
       AND percentile = 1
ORDER BY total_price desc;
```

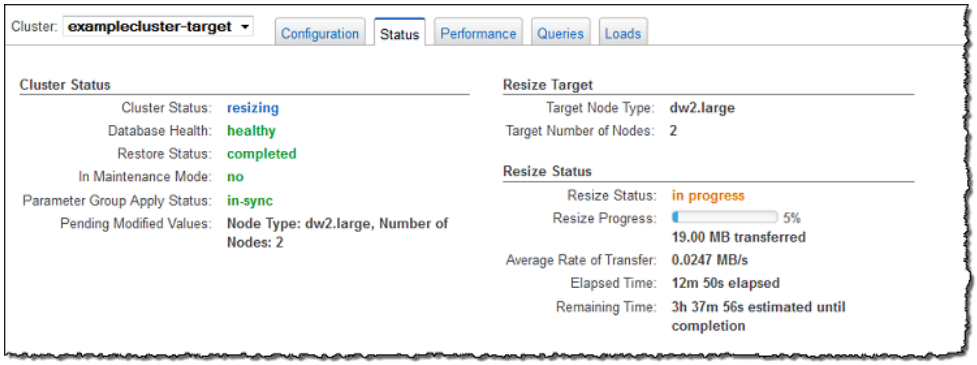
Step 4: Resize the Target Cluster

Once you verify that your target cluster works as expected, you can resize the target cluster. You can continue to allow write and read-write operations in the source cluster, because later in this tutorial you will copy any data that was loaded after your snapshot to the target.

1. Open the Amazon Redshift console.
2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from this tutorial, click **examplecluster-target**.
3. On the **Configuration** tab of the **Cluster** details page, click **Resize** in the **Cluster** list.
4. In the **Resize Cluster** window, select the following values:
   - **Node Type**: **dw2.large**.
   - **Cluster Type**: **Multi Node**.
   - **Number of Nodes**: **2**.


Resize Cluster
Choose the number of nodes and optionally a new node type for the resize operation. Note that the available node type and cluster type options may be limited by the cluster's current availability zone.
Node Type
dw2.large
Cluster Type
Multi Node
Number Of Nodes*
2
Please make sure the resized cluster is large enough to hold the data that is currently on the cluster; otherwise the resize will fail.
Note: Resizing the cluster will cause it to be restarted into read-only mode for the duration of the resize operation. All currently executing queries and database connections on the cluster will be terminated when the resize operation begins and again when it is complete.
Cancel
Resize

5. Click **Resize**.
6. Click **Status**, and review the resize status information to see the resize progress.



Step 5: Copy Post-Snapshot Data from the Source to the Target Cluster

For the purposes of this tutorial, this step provides a simple set of COPY statements to load data from Amazon S3 into Amazon Redshift. This step is included to simulate bringing the target cluster up-to-date with the same data as the source cluster. It is not meant to demonstrate an effort to bring an actual production environment into line between the source and target cluster. In production environments, your own ETL process will determine how load your target cluster with all the same data as the source cluster after the snapshot was taken.

If there have been multiple loads after the snapshot was taken, you'll need to make sure that you rerun the loads in the target database in the same order as they were run in the source database. Additionally, if there continue to be loads into the source database while you are working on bringing the target cluster up-to-date, you will need to repeat this process until the target and source match, and find a suitable time to rename the clusters and switch applications to connect to the target database.

In this example, let's suppose that your ETL process loaded data into the source cluster after the snapshot was taken. Perhaps Amazon Redshift was still in the process of restoring the target cluster from the snapshot, or resizing the target cluster. There were some new categories, events, dates, and venues added to the TICKIT database. You now need to get this same data into the target cluster before you switch to use it going forward.

First, you'll use the following COPY statements to load new data from Amazon S3 to the tables in your Amazon Redshift TICKIT database in the target cluster.

```
copy venue from 's3://<region-specific-bucket-name>/resize/etl_venue_pipe.txt'
 CREDENTIALS 'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-
Secret-Access-Key>' delimiter '|';
copy category from 's3://<region-specific-bucket-name>/resize/etl_cat
egory_pipe.txt' CREDENTIALS 'aws_access_key_id=<Your-Access-Key-
ID>;aws_secret_access_key=<Your-Secret-Access-Key>' delimiter '|';
copy date from 's3://<region-specific-bucket-name>/resize/etl_date_pipe.txt'
CREDENTIALS 'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-
Secret-Access-Key>' delimiter '|';
copy event from 's3://<region-specific-bucket-name>/resize/etl_events_pipe.txt'
 CREDENTIALS 'aws_access_key_id=<Your-Access-Key-ID>;aws_secret_access_key=<Your-
Secret-Access-Key>' delimiter '|'  timeformat 'YYYY-MM-DD HH:MI:SS';
```

You must replace *<Your-Access-Key-ID>* and *<Your-Secret-Access-Key>* with your own credentials and *<region-specific-bucket-name>* with the name of a bucket in the same region as your cluster. We recommend that you use temporary credentials to perform this COPY operation. For more information about using temporary credentials, see Temporary security credentials in the *Amazon Redshift Database Developer Guide*. Use the following table to find the correct bucket name to use.

Region	*<region-specific-bucket-name>*
US East (N. Virginia)	awssampledb
US West (Oregon)	awssampledbuswest2
EU (Frankfurt)	awssampledbeucentral1
EU (Ireland)	awssampledbeuwest1
Asia Pacific (Singapore)	awssampledbapsoutheast1
Asia Pacific (Sydney)	awssampledbapsoutheast2
Asia Pacific (Tokyo)	awssampledbapnortheast1

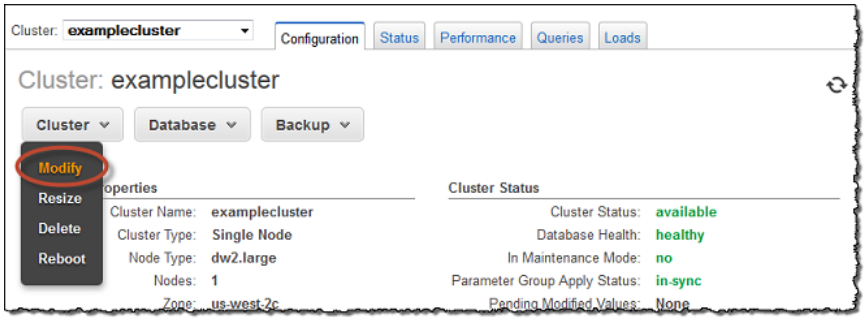
Note
In this exercise, you upload sample data from existing Amazon S3 buckets, which are owned by Amazon Redshift. The bucket permissions are configured to allow everyone read access to the sample data files. If you want to upload your own data, you must have your own Amazon S3 bucket. For information about creating a bucket and uploading data, go to Creating a Bucket and Uploading Objects into Amazon S3 in the *Amazon Simple Storage Service Console User Guide*.

Step 6: Rename the Source and Target Clusters

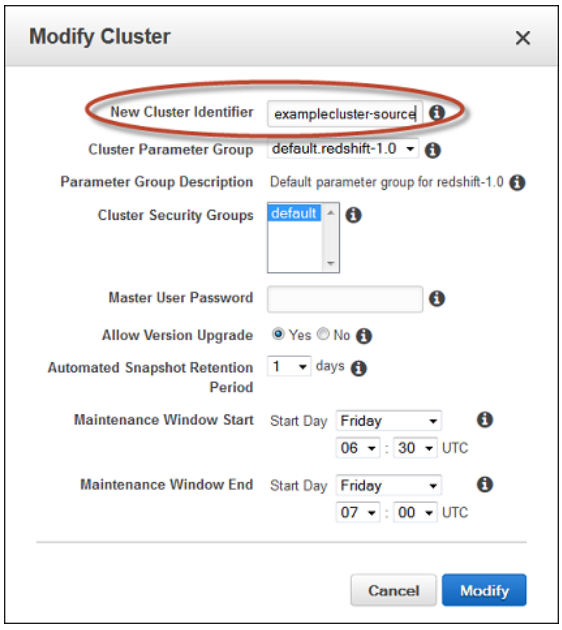
Once you verify that your target cluster has been brought up to date with any data needed from the ETL process, you can switch to the target cluster. If you need to keep the same name as the source cluster, you'll need to do a few manual steps to make the switch. These steps involve renaming the source and target clusters, during which time they will be unavailable for a short period of time. However, if you are able to update any data sources to use the new target cluster, you can skip this section.

1. Open the Amazon Redshift console.

2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster from this tutorial, click **examplecluster**.
3. On the **Configuration** tab of the **Cluster** details page, click **Modify** in the **Cluster** list.



4. In the **Modify Cluster** window, type **examplecluster-source** in the **New Cluster Identifier** box, and then click **Modify**.



5. In the navigation pane, click **Clusters**, and then click **examplecluster-target**.
6. On the **Configuration** tab of the **Cluster** details page, click **Modify** in the **Cluster** list.
7. In the **Modify Cluster** window, type **examplecluster** in the **New Cluster Identifier** box, and then click **Modify**.

If you had any queries running in the source cluster, you'll need to start them over and run them to completion on the target cluster.

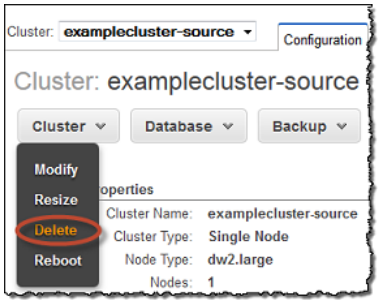
Step 7: Delete the Source Cluster

After you are sure that you no longer need the source cluster, you can delete it. In a production environment, whether you decide to keep a final snapshot depends on your data policies. In this tutorial, you'll delete the cluster without a final snapshot because you are using sample data.

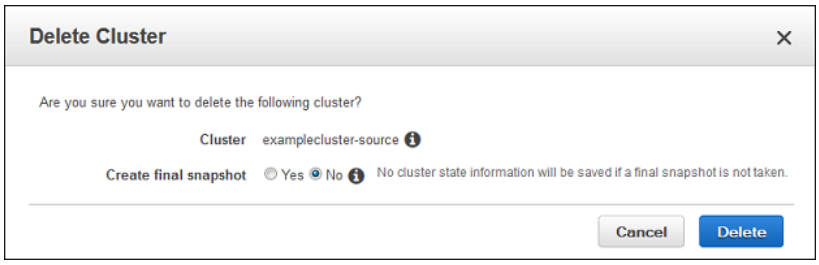
Important
You are charged for any clusters until they are deleted.

1. Open the Amazon Redshift console.

2. In the navigation pane, click **Clusters**, and then click the cluster to open. If you are using the same cluster names from this tutorial, click **examplecluster-source**.
3. On the **Configuration** tab of the **Cluster** details page, click **Delete** in the **Cluster** list.



4. In the **Delete Cluster** window, click **No** for **Create final snapshot**, and then click **Delete**.


Delete Cluster
Are you sure you want to delete the following cluster?
Cluster examplecluster-source
Create final snapshot Yes No
No cluster state information will be saved if a final snapshot is not taken.
Cancel
Delete

Step 8: Clean Up Your Environment

After you have completed this tutorial, you can clean up your environment by deleting the target cluster. To do this, follow the steps in Step 7: Delete the Source Cluster (p. 235) and instead delete the target cluster. Doing this will return your environment back to the state it was in before you started the tutorial. Returning the environment to the original state is important to help reduce any costs associated with having clusters running.

Important
You are charged for any clusters until they are deleted.

Limits in Amazon Redshift

Topics

- Quotas and Limits (p. 237)
- Naming Constraints (p. 238)

Quotas and Limits

Amazon Redshift has quotas that limit the number of clusters you can create, the total number of nodes that you can provision, and the number of snapshots that you can create; these quotas are per AWS account per region. Amazon Redshift has a default quota for each of these, which are listed at AWS Service Limits. If you attempt to exceed any of these quotas, the attempt will fail. To increase these Amazon Redshift quota limits for your account in a region, request a change by submitting an Amazon Redshift Limit Increase Form.

In addition to quotas, Amazon Redshift has limits for the following per-cluster values. These limits cannot be increased:

- The number of nodes that you can allocate per cluster, which is based on the cluster's node type. This limit is separate from the limit for your AWS account per region. For more information about the current node limits for each node type, see About Clusters and Nodes (p. 6).
- The number of permanent tables that you can create per cluster is 9,990. For more information about creating a table, go to Create Table Usage Notes in the *Amazon Redshift Database Developer Guide*.
- The number of user-defined databases you can create per cluster is 60. For more information about creating a database, go to Create Database in the *Amazon Redshift Database Developer Guide*.
- The number of schemas you can create per database is 256. For more information about creating a schema, go to Create Schema in the *Amazon Redshift Database Developer Guide*.
- The number of concurrent user connections that can be made to a cluster is 500. For more information, see Connecting to a Cluster (p. 138) in the *Amazon Redshift Cluster Management Guide*.
- The number of AWS accounts you can authorize to restore a snapshot is 20. For more information, see Sharing Snapshots (p. 84) in the *Amazon Redshift Cluster Management Guide*.
- The maximum size of a single row loaded by using the COPY command is 4 MB. For more information, go to COPY in the *Amazon Redshift Database Developer Guide*.

Naming Constraints

The following table describes naming constraints within Amazon Redshift.

Cluster identifier	• A cluster identifier must contain only lowercase characters. • It must contain from 1 to 63 alphanumeric characters or hyphens. • Its first character must be a letter. • It cannot end with a hyphen or contain two consecutive hyphens. • It must be unique for all clusters within an AWS account.
Database name	• A database name must contain 1 to 64 alphanumeric characters. • It must contain only lowercase letters. • It cannot be a reserved word. For a list of reserved words, to Reserved Words in the *Amazon Redshift Database Developer Guide*.
Master user name	• A master user name must contain only lowercase characters. • It must contain from 1 to 128 alphanumeric characters. • Its first character must be a letter. • It cannot be a reserved word. For a list of reserved words, go to Reserved Words in the *Amazon Redshift Database Developer Guide*.
Master password	• A master password must be between 8 and 64 characters in length. • It must contain at least one uppercase letter. • It must contain at least one lowercase letter. • It must contain one number. • It can be any printable ASCII character (ASCII code 33 to 126) except ' (single quotation mark), " (double quotation mark), \, /, @, or space.
Parameter group name	• A parameter group name must be 1 to 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • Its first character must be a letter. • It cannot end with a hyphen or contain two consecutive hyphens.

Cluster security group name	• A cluster security group name must contain no more than 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • It must not be **`Default`**. • It must be unique for all security groups that are created by your AWS account.
Subnet group name	• A subnet group name must contain no more than 255 alpha-numeric characters or hyphens. • It must contain only lowercase characters. • It must not be **`Default`**. • It must be unique for all security groups that are created by your AWS account.
Cluster snapshot identifier	• A cluster snapshot identifier must contain no more than 255 alphanumeric characters or hyphens. • It must contain only lowercase characters. • It must not be **`Default`**. • It must be unique for all security groups that are created by your AWS account.

Tagging Resources in Amazon Redshift

Topics

- Tagging Overview (p. 240)
- Managing Resource Tags Using the Console (p. 241)
- Managing Tags Using the Amazon Redshift API (p. 243)

Tagging Overview

In AWS, tags are user-defined labels that consist of key-value pairs. Amazon Redshift supports tagging to provide metadata about resources at a glance, and to categorize your billing reports based on cost allocation. To use tags for cost allocation, you must first activate those tags in the AWS Billing and Cost Management service. For more information about setting up and using tags for billing purposes, see Use Cost Allocation Tags for Custom Billing Reports and Setting Up Your Monthly Cost Allocation Report.

Tags are not required for resources in Amazon Redshift, but they help provide context. You might want to tag resources with metadata about cost centers, project names, and other pertinent information related to the resource. For example, suppose you want to track which resources belong to a test environment and a production environment. You could create a key named `environment` and provide the value `test` or `production` to identify the resources used in each environment. If you use tagging in other AWS services or have standard categories for your business, we recommend that you create the same key-value pairs for resources in Amazon Redshift for consistency.

Tags are retained for resources after you resize a cluster, and after you restore a snapshot of a cluster within the same region. However, tags are not retained if you copy a snapshot to another region, so you must recreate the tags in the new region. If you delete a resource, any associated tags are deleted.

Each resource has one *tag set*, which is a collection of one or more tags assigned to the resource. Each resource can have up to 10 tags per tag set. You can add tags when you create a resource and after a resource has been created. You can add tags to the following resource types in Amazon Redshift:

- CIDR/IP
- Cluster
- Cluster security group
- Cluster security group ingress rule

- EC2 security group
- HSM connection
- HSM client certificate
- Parameter group
- Snapshot
- Subnet group

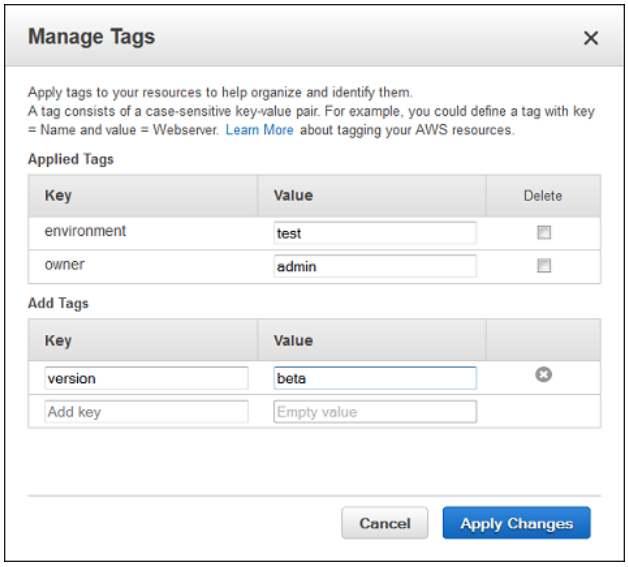
Tagging Requirements

Tags have the following requirements:

- Keys can't be prefixed with `aws:`.
- Keys must be unique per tag set.
- A key must be between 1 and 128 allowed characters.
- A value must be between 0 and 256 allowed characters.
- Values do not need to be unique per tag set.
- Allowed characters for keys and values are Unicode letters, digits, white space, and any of the following symbols: _ . : / = + - @.
- Keys and values are case sensitive.

Managing Resource Tags Using the Console

The following is an example of the **Manage Tags** window for an Amazon Redshift resource, such as a cluster or a parameter group.

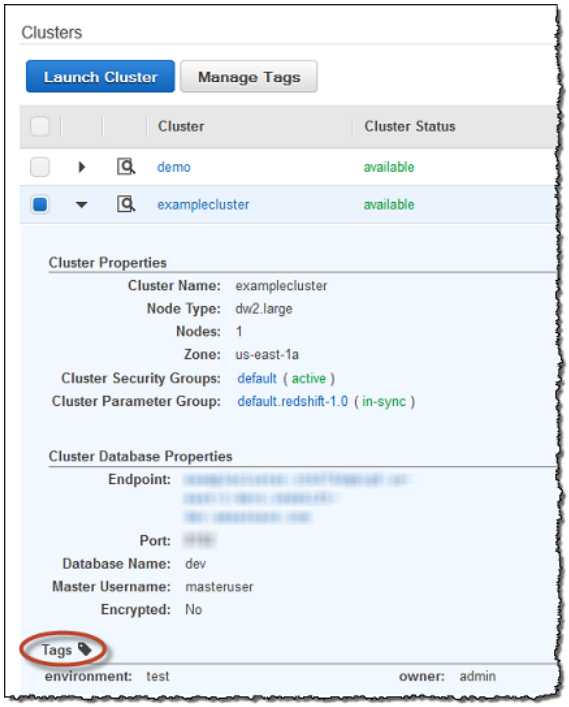


You use the **Add Tags** section to add key pairs to an Amazon Redshift resource. When you begin entering a key pair in the **Add Tags** section, a new row will appear so that you can add another key pair, and so on. For more information about allowed characters for keys and values, see Tagging Requirements (p. 241).

If you decide that you don't want to add a particular tag to the resource, you can remove it from the **Add Tags** section by clicking the **X** in the row. Once you have specified the key pairs that you want to add, you apply the changes so that they are associated with the resource.

After you add key pairs to a resource, they display in the **Applied Tags** section; this is the tag set for the resource. You can modify a tag value, but you can't modify the key name. You can, however, delete a key if you no longer need it for the resource.

You can view the tags for a resource by reviewing the **Applied Tags** section of the **Manage Tags** window. Alternatively, you can quickly view tags by navigating to a resource type in the navigation pane, and then expanding the resource in the list to view the **Tags** section. The following is an example of a cluster expanded to show various properties, including tags associated with the cluster.



How To Open the Manage Tags Window

The following table describes how to open the **Manage Tags** window for each of the Amazon Redshift resources that support tags.

Resource	Description
Cluster	In the left navigation pane, click **Clusters** and select a cluster from the list. Then click **Manage Tags**.
Snapshot	In the left navigation pane, click **Snapshots** and select a snapshot from the list. Then click **Actions**, and click **Manage Tags** from the menu.
Cluster Security Group	In the left navigation pane, click **Security**. On the **Security Groups** tab, select a security group from the list. Then click **Manage Tags**.
Cluster Security Group Ingress Rule	In the left navigation pane, click **Security**. On the **Security Groups** tab, click a security group in the list. On the **Security Group Connections** page, select an ingress rule and then click **Manage Tags**.
Subnet Group	In the left navigation pane, click **Security**. On the **Subnet Groups** tab, select a subnet group from the list. Then click **Manage Tags**.
HSM Connection	In the left navigation pane, click **Security**. On the **HSM Connections** tab, select a connection from the list. Then click **Manage Tags**.

Resource	Description
HSM Certificate	In the left navigation pane, click **Security**. On the **HSM Certificates** tab, select a certificate from the list. Then click **Manage Tags**.
Parameter Group	In the left navigation pane, click **Parameter Groups** and select a parameter group from the list. Then click **Manage Tags**.

How to Manage Tags in the Amazon Redshift Console

Use the table in the previous section to navigate to the resource that you want to work with, and then use the procedures in this section to add, modify, delete, and view tags for the resource.

To add tags to a resource

1. Navigate to the resource to which you want to add tags, and open the **Manage Tags** window.
2. Under **Add Tags**, type a key name in the **Key** box and the key value in the **Value** box. For example, type `environment` in the **Key** box and `production` in the **Value** box. Repeat this step to add any additional tags.
3. Click **Apply Changes**.

To modify tags associated with a resource

1. Navigate to the resource for which you want to modify tags, and open the **Manage Tags** window.
2. Under **Applied Tags**, locate the key that you want to modify. In the **Value** box, type a new key value. Repeat for any other tags that you want to modify.
3. Click **Apply Changes**.

To delete tags associated with a resource

1. Navigate to the resource from which you want to delete tags, and open the **Manage Tags** window.
2. Under **Applied Tags**, locate the key that you want to delete. Select the **Delete** check box. Repeat for any other tags that you want to delete.
3. Click **Apply Changes**.

Managing Tags Using the Amazon Redshift API

You can use the following Amazon Redshift APIs to manage tags:

- CreateTags
- DeleteTags
- DescribeTags
- Tag
- TaggedResource

Additionally, you can use the following Amazon Redshift APIs to manage and view tags for a specific resource:

- CreateCluster
- CreateClusterParameterGroup
- CreateClusterSecurityGroup
- CreateClusterSnapshot
- CreateClusterSubnetGroup
- CreateHsmClientCertificate
- CreateHsmConfiguration
- DescribeClusters
- DescribeClusterParameterGroups
- DescribeClusterSecurityGroups
- DescribeClusterSnapshots
- DescribeClusterSubnetGroups
- DescribeHsmClientCertificates
- DescribeHsmConfigurations

Document History

The following table describes the important changes to the *Amazon Redshift Management Guide*.

Latest documentation update: April 30, 2015

For a list of the changes to the Amazon Redshift database documentation, go to the Amazon Redshift Developer Guide Document History.

Change	Description	Release Date
Documentation Fixes	Published various documentation fixes.	April 30, 2015
Documentation Update	Updated the download links to new versions of the Amazon Redshift JDBC drivers, and added a section for previous versions of these drivers. Also added a link to release notes for the drivers. For more information, see Configure a JDBC Connection (p. 140).	April 1, 2015
Documentation Update	Added downloads for new versions of the Amazon Redshift JDBC drivers. Also updated the format of the Amazon Redshift JDBC URL. For more information, see Configure a JDBC Connection (p. 140). Added cluster security group ingress rules as a taggable resource. For more information, see Tagging Resources in Amazon Redshift (p. 240). Updated the instructions for adding a cluster security group ingress rule, and added instructions for tagging a cluster security group ingress rule. For more information, see Managing Cluster Security Groups Using the Console (p. 44).	March 16, 2015
New Feature	This release of Amazon Redshift introduces new ODBC and JDBC drivers optimized for use with Amazon Redshift. For more information, see Connecting to a Cluster (p. 138).	February 26, 2015
New Feature	This release of Amazon Redshift introduces cluster performance metrics that allow you to view and analyze query execution details. For more information, see Viewing Query Performance Data (p. 186).	February 26, 2015

Change	Description	Release Date
Documentation Update	Added a troubleshooting topic about an issue where queries fail to complete in the SQL client tool and the cluster when a transaction encounters an error. For more information, see Queries Appear to Hang in the Client and the Cluster (p. 177).	January 26, 2015
Documentation Update	Added a new example policy that demonstrates granting permission to common AWS service actions and resources on which Amazon Redshift relies. For more information, see Example Policies for Amazon Redshift (p. 121).	January 16, 2015
Documentation Update	Updated guidance about setting the maximum transmission unit (MTU) to disable TCP/IP jumbo frames. For more information, see Supported Platforms to Launch Your Cluster (p. 9) and Queries Appear to Hang in the Client and Do Not Reach the Cluster (p. 176).	January 16, 2015
Documentation Update	Revised the content about the *wlm_json_configuration* parameter, and provided example syntax to configure this parameter by using the AWS CLI on the Linux, Mac OS X, and Microsoft Windows operating systems. For more information, see Configuring Workload Management (p. 59).	January 13, 2015
Documentation Update	Added missing event notifications and descriptions. For more information, see Amazon Redshift Event Categories and Event Messages (p. 204).	January 8, 2015
Documentation Update	Updated guidance about IAM policies for Amazon Redshift actions and resources. Revised the section to improve organization and clarity. For more information, see Controlling Access to Amazon Redshift Resources (p. 115).	November 21, 2014
New Feature	This release of Amazon Redshift introduces the ability to encrypt clusters using encryption keys from AWS Key Management Service (AWS KMS). AWS KMS combines secure, highly available hardware and software to provide a key management system scaled for the cloud. For more information about AWS KMS and encryption options for Amazon Redshift, see Amazon Redshift Database Encryption (p. 98) and Managing Clusters Using the Console (p. 13).	November 12, 2014
New Feature	This release of Amazon Redshift introduces the ability to tag resources, such as clusters and snapshots. Tags enable you to provide user-defined metadata to categorize your billing reports based on cost allocation, and to help you better identify resources at a glance. For more information, see Tagging Resources in Amazon Redshift (p. 240).	November 4, 2014
New Feature	Increased the maximum node limit to 128 nodes for dw1.8xlarge and dw2.8xlarge node sizes. For more information, see About Clusters and Nodes (p. 6).	October 30, 2014

Change	Description	Release Date
Documentation Update	Added links to the Microsoft Visual C++ 2010 Redistributable Packages that are required for Amazon Redshift to use PostgreSQL ODBC drivers. For more information, see Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems (p. 146).	October 30, 2014
New Feature	Added the ability to terminate queries and loads from the Amazon Redshift console. For more information, see Viewing Query Performance Data (p. 186) and Viewing Cluster Metrics During Load Operations (p. 194).	October 28, 2014
Documentation Fixes	Published various documentation fixes.	October 17, 2014
New Content	Added content about shutting down clusters and deleting clusters. For more information, see Shutting Down and Deleting Clusters (p. 12) and Shutting Down or Deleting a Cluster (p. 22).	August 14, 2014
Documentation Update	Clarified the behavior of the **Allow Version Upgrade** setting for clusters. For more information, see Overview (p. 5).	August 14, 2014
Documentation Update	Revised procedures, screenshots, and organization of topic about working with clusters in Amazon Redshift console. For more information, see Managing Clusters Using the Console (p. 13).	July 11, 2014
New Content	Added a new tutorial about resizing Amazon Redshift clusters, including how to resize a cluster while minimizing the amount of time that the cluster is in read-only mode. For more information, see Tutorial: Resizing Clusters in Amazon Redshift (p. 224).	June 27, 2014
New Feature	Added the ability to rename clusters. For more information, see Renaming Clusters (p. 11) and Modifying a Cluster (p. 20).	June 2, 2014
Documentation Update	Updated the .NET code example to use the ODBC data provider when connecting to a cluster programmatically by using .NET. For more information, see Connecting to a Cluster by Using .NET (p. 171).	May 15, 2014
New Feature	Added options to select a different parameter group and security group when you restore a cluster from a snapshot. For more information, see Restoring a Cluster from a Snapshot (p. 89).	May 12, 2014
New Feature	Added new section to describe how to configure a default Amazon CloudWatch alarm to monitor the percentage of disk space used in an Amazon Redshift cluster. This alarm is a new option in the cluster creation process. For more information, see Default Disk Space Alarm (p. 11).	April 28, 2014
Documentation Update	Clarified information about Elliptic curve Diffie—Hellman Exchange (ECDHE) support in Amazon Redshift. For more information, see Connect Using SSL (p. 158).	April 22, 2014

Change	Description	Release Date
New Feature	Added statement about Amazon Redshift support for the Elliptic curve Diffie—Hellman (ECDH) key agreement protocol. For more information, see Connect Using SSL (p. 158).	April 18, 2014
Documentation Update	Revised and reorganized the topics in the Connecting to a Cluster (p. 138) section. Added more information about JDBC and ODBC connections, and a new troubleshooting section for connection issues.	April 15, 2014
Documentation Update	Added version in IAM policy examples throughout the guide.	April 3, 2014
Documentation Update	Added information about how pricing works when you resize a cluster. For more information, see Purchasing Amazon Redshift Reserved Nodes (p. 107).	April 2, 2014
New Feature	Added a section about a new parameter, *max_cursor_result_set_size*, which sets the maximum result set size, in megabytes, that can be stored per individual cursor. This parameter value also affects the number of concurrently active cursors for the cluster. For more information, see Amazon Redshift Parameter Groups (p. 56).	March 28, 2014
New Feature	Added explanation about the **Cluster Version** field now including both cluster engine version and database revision number. For more information, see Amazon Redshift Clusters (p. 5).	March 21, 2014
New Feature	Updated the resize procedure to show the new resize progress information on the cluster's **Status** tab. For more information, see Resizing a Cluster (p. 24).	March 21, 2014
Documentation Update	Reorganized and updated What Is Amazon Redshift? (p. 1) and revised Amazon Redshift Management Overview (p. 2). Published various documentation fixes.	February 21, 2014
New Feature	Added new node types and sizes for Amazon Redshift clusters, and rewrote the related cluster overview topic for better organization and clarity based on feedback. For more information, see Amazon Redshift Clusters (p. 5).	January 23, 2014
New Feature	Added information about using elastic IP (EIP) addresses for publicly-accessible Amazon Redshift clusters in virtual private clouds. For more information about EIP in Amazon Redshift, see Managing Clusters in an Amazon Virtual Private Cloud (VPC) (p. 32) and Creating a Cluster in a VPC (p. 33).	December 20, 2013
New Feature	Added information about the AWS CloudTrail logs for Amazon Redshift. For more information about Amazon Redshift support for CloudTrail, see Using AWS CloudTrail for Amazon Redshift (p. 221).	December 13, 2013

Change	Description	Release Date
New Feature	Added information about the new user activity log and the *enable_user_activity_logging* database parameter for the database audit logging feature in Amazon Redshift. For more information about database audit logging, see Database Audit Logging (p. 215). For more information about database parameters, see Amazon Redshift Parameter Groups (p. 56).	December 6, 2013
New Feature	Updated to describe configuring Amazon Redshift to automatically copy automated and manual snapshots to a secondary region. For more information about configuring cross-region snapshot copy, see Copying Snapshots to Another Region (p. 83).	November 14, 2013
New Feature	Added section to describe Amazon Redshift audit logging for connection and user activity, and storing these logs in Amazon S3. For more information about database audit logging, see Database Audit Logging (p. 215).	November 11, 2013
New Feature	Added section to describe Amazon Redshift encryption with new features for managing encryption keys in a hardware security module (HSM) and rotating encryption keys. For more information about encryption, HSM, and key rotation, see Amazon Redshift Database Encryption (p. 98), Hardware Security Modules (p. 99), and Rotating Encryption Keys (p. 105).	November 11, 2013
New Feature	Updated to describe publishing notifications of Amazon Redshift events by using Amazon SNS. For information about Amazon Redshift event notifications, see Amazon Redshift Event Notifications (p. 202).	November 11, 2013
New Feature	Updated to describe IAM resource level permissions. For information about Amazon Redshift IAM permissions, see Controlling Access to Amazon Redshift Resources (p. 115).	August 9, 2013
New Feature	Updated to describe restore progress metrics. For more information, see Restoring a Cluster from a Snapshot (p. 84).	August 9, 2013
New Feature	Updated to describe cluster snapshot sharing and create snapshot progress metrics. For more information, see Sharing Snapshots (p. 84).	July 17, 2013
Documentation Fixes	Published various documentation fixes.	July 8, 2013
New Console Screens	Updated the *Amazon Redshift Management Guide* to match changes in the Amazon Redshift console.	April 22, 2013
New Guide	This is the first release of the *Amazon Redshift Management Guide*.	February 14, 2013